

NASA/CR-2001-210872  
ICASE Report No. 2001-14



## **Global Discrete Artificial Boundary Conditions for Time-dependent Wave Propagation**

*V.S. Ryaben'kii*

*Keldysh Institute for Applied Mathematics, Moscow, Russia*

*S.V. Tsynkov*

*North Carolina State University, Raleigh, North Carolina and  
Tel Aviv University, Tel Aviv, Israel*

*V.I. Turchaninov*

*Keldysh Institute for Applied Mathematics, Moscow, Russia*

*ICASE*

*NASA Langley Research Center, Hampton, Virginia*

*Operated by Universities Space Research Association*



National Aeronautics and  
Space Administration

Langley Research Center  
Hampton, Virginia 23681-2199

**DISTRIBUTION STATEMENT A**  
**Approved for Public Release**  
**Distribution Unlimited**

Prepared for Langley Research Center  
under Contract NAS1-97046

May 2001

20010822 153

# GLOBAL DISCRETE ARTIFICIAL BOUNDARY CONDITIONS FOR TIME-DEPENDENT WAVE PROPAGATION\*

V. S. RYABEN'KII<sup>†</sup>, S. V. TSYNKOV<sup>§¶</sup>, AND V. I. TURCHANINOV<sup>||</sup>

**Abstract.** We construct global artificial boundary conditions (ABCs) for the numerical simulation of wave processes on unbounded domains using a special non-deteriorating algorithm that has been developed previously for the long-term computation of wave-radiation solutions. The ABCs are obtained directly for the discrete formulation of the problem; in so doing, neither a rational approximation of "non-reflecting kernels," nor discretization of the continuous boundary conditions is required. The extent of temporal nonlocality of the new ABCs appears fixed and limited; in addition, the ABCs can handle artificial boundaries of irregular shape on regular grids with no fitting/adaptation needed and no accuracy loss induced.

The non-deteriorating algorithm, which is the core of the new ABCs, is inherently three-dimensional, it guarantees temporally uniform grid convergence of the solution driven by a continuously operating source on arbitrarily long time intervals, and provides unimprovable linear computational complexity with respect to the grid dimension. The algorithm is based on the presence of lacunae, i.e., aft fronts of the waves, in wave-type solutions in odd-dimension spaces. It can, in fact, be built as a modification on top of any consistent and stable finite-difference scheme, making its grid convergence uniform in time and at the same time keeping the rate of convergence the same as that of the non-modified scheme.

In the paper, we delineate the construction of the global lacunae-based ABCs in the framework of a discretized wave equation. The ABCs are obtained for the most general formulation of the problem that involves radiation of waves by moving sources (e.g., radiation of acoustic waves by a maneuvering aircraft). We also present systematic numerical results that corroborate the theoretical design properties of the ABCs' algorithm.

**Key words.** artificial boundary conditions, wave propagation, lacunae, non-deteriorating method

**Subject classification.** Applied and Numerical Mathematics

**1. Introduction.** Numerical simulation of wave phenomena on unbounded domains (e.g., radiation and/or scattering of acoustic or electromagnetic waves), often encounters the following well-recognized difficulty. As no computer can either handle infinite arrays of data or perform infinite numbers of arithmetics operations, the discrete approximation of the problem has to be made finite (i.e., finite-dimensional). Consequently, the original infinite domain has to be truncated and special artificial boundary conditions (ABCs) have to be developed as a closure for the resulting finite formulation.

The literature on the subject of ABCs is very extensive, see, e.g., review papers by Givoli [1] and

\* The work of the first and second authors was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046, and by Director's Discretionary Fund, NASA Langley Research Center, while they were in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681-2199, USA.

<sup>†</sup>Keldysh Institute for Applied Mathematics, Russian Ac. Sci., 4 Miusskaya Sq., Moscow 125047, Russia; and ICASE, MS 132C, NASA Langley Research Center, Hampton, VA 23681-2199, USA.

<sup>§</sup>Corresponding author. Department of Mathematics, North Carolina State University, Box 8205, Raleigh, NC 27695, USA. Phone: (1-919)515-1877, Fax: (1-919)515-3798. E-mail: [tsynkov@math.ncsu.edu](mailto:tsynkov@math.ncsu.edu), URL: [www.math.ncsu.edu/~stsynkov](http://www.math.ncsu.edu/~stsynkov).

<sup>¶</sup>Also: School of Mathematical Sciences, Tel Aviv University, Ramat Aviv, Tel Aviv 69978, Israel; and ICASE, MS 132C, NASA Langley Research Center, Hampton, VA 23681-2199, USA.

<sup>||</sup>Keldysh Institute for Applied Mathematics, Russian Ac. Sci., 4 Miusskaya Sq., Moscow 125047, Russia.

Tsynkov [2], as well as another recent review by Hagstrom [3], which is geared more specifically toward wave propagation problems. In the current study, we are going to focus on genuinely unsteady (as opposed to time-harmonic) wave phenomena to be computed in the time domain. For this type of problems, most of the ABCs' research to date has been done in the framework of simple approximate local methods based, e.g., on quasi-one-dimensional characteristics' decomposition. These methods often appear insufficiently accurate. Some of the more accurate methods that have recently gained attention are based on the so-called perfectly matched layers (PML), see the original publications [4–7] and reviews [8, 9]. Unfortunately, as shown in [10, 11], these methods may give rise to instabilities of a particular kind. The latter typically manifest themselves when integrating over long time intervals and thus exacerbate even further the well-known problem of accumulation of error in long-term numerical simulations.

Among other existing unsteady ABCs' approaches, only very few methodologies can guarantee the accuracy that theoretically would not hamper that of the interior approximation. All of these methodologies are non-local, see, e.g., [12–18], which is characteristic of highly-accurate (ideally, exact) ABCs. The techniques of this group typically involve two “approximating” steps, which are undertaken consecutively when building the ABCs. The first step is the replacement of the fully non-local in space-time true exact boundary conditions, which are most often written using pseudo-differential operators, with approximate boundary conditions (still at the continuous level) that would provide for only a limited extent of non-locality. More precisely, this step aims at limiting the temporal nonlocality of the ABCs, which may be prohibitively expensive in computations. This can be achieved, e.g., by employing a rational approximation of the corresponding symbol (kernel).<sup>1</sup> The first step is then followed by the second one, which is the discretization of the resulting localized continuous boundary conditions. We note that unless given a special thorough attention, the discretization step may lead to accuracy loss and again cause instability (this pertains to purely local ABCs as well). We also note that all of these techniques are restricted geometrically to computational domains of a simple regular shape, e.g., those with spherical or planar boundaries.

Our recent work [22, 23] indicates that the issue of time-dependent ABCs may be closely related to another problem that has been mentioned before — the accumulation of numerical error during long runs. This problem has been recognized as an outstanding question in computational PDEs for many years, since the first systematic convergence studies for discrete approximations have been conducted in the fifties. From the standpoint of practical computing, deterioration of the solution over long time intervals can be attributed, e.g., to the mechanism of either numerical dissipation or dispersion or both. Theoretically, this phenomenon is conveniently termed as non-uniformity of the grid convergence in time, and all conventional discrete approximations that are currently in use are known to suffer from this deficiency.

As our work [22, 23] suggests, the key to resolving the question of long-term error accumulation may be in precisely following the physical nature of the original problem when building a numerical algorithm. Namely, it is known that waves in three dimensions have aft (or trailing) fronts. This is a manifestation of the so-called Huygens' principle or more generally, the presence of lacunae in wave-type solutions in odd-dimension spaces. Using this property, we have been able to develop a modification to any consistent and stable finite-difference scheme that approximates a Cauchy problem for the wave equation so as to make its grid convergence uniform in time on arbitrarily long intervals. The uniform temporal convergence of the algorithm has been proven theoretically along with its optimal computational complexity (i.e., linear with

<sup>1</sup>In fact, a wide variety of purely local ABCs (i.e., local in both space and time) can be obtained via rational approximation of symbols as well; this approach has been known for two decades, see [19–21]. The general trend is that the more of the nonlocal nature of exact ABCs is compromised, the less accuracy one can expect from the resulting approximate methodology.

respect to the grid dimension). The rate of temporally uniform grid convergence, see [22, 23], remains the same as that of the original non-modified scheme. These results apply to the general case of moving sources of waves that operate continuously in time. As an example, we show in [23] that the linearized flow around a maneuvering aircraft can be interpreted in this framework.

At least as important, the procedure of [22, 23] allows one to replace the original infinite domain of the initial-value problem by a finite computational domain that would facilitate the construction of a finite-dimensional discretization. As will be seen from the discussion in the current paper, the latter replacement leads to obtaining highly accurate non-local unsteady ABCs for combined problems that may include complex phenomena on a bounded interior domain but reduce to the homogeneous wave equation in the far field. Similarly to the case analyzed in [22, 23] the interior domain may be moving. The ABCs are built directly for the specific interior approximation used and in this sense can be considered its most natural extension. We emphasize that they involve neither of the two common approximating steps (rational approximation of the symbol followed by discretization) that have been mentioned before in connection to some existing ABCs' methodologies.

Unlike in all other methods available in the literature, the extent of temporal nonlocality of the unsteady ABCs based on the technique [22, 23] is bounded naturally for all times because of the explicit use of lacunae. This bound is not a consequence of any approximation. Moreover, as these ABCs are obtained directly for the specific finite-difference scheme, the issue of discretization of the boundary conditions, which has been shown to cause problems before, simply does not arise in this framework. Besides, the new ABCs possess full geometric universality, i.e., can handle any shape of the external artificial boundary with equal ease on a regular Cartesian grid with no fitting/adaptation required and no accuracy loss caused.

The rest of the paper is organized as follows. In Section 2, we provide for a brief outline of the phenomenon of lacunae in wave radiation solutions, and show how one can use those to obtain a non-deteriorating algorithm for the long-term numerical integration of the corresponding problems. In addition to the theoretical justification, we also include in this section several computational demonstrations of the properties of the aforementioned algorithm. In Section 3, we describe in detail the construction of the global finite-difference lacunae-based ABCs, and briefly comment on how the proposed construction fits into the general framework of discrete time-dependent boundary conditions developed by Ryaben'kii in [24]. Finally, Section 4 contains an extensive set of numerical experiments with the new ABCs for the wave equation. The experiments are conducted for finite-difference schemes of different orders of accuracy, different laws of motion for the waves' sources (uniform, as well as non-uniform), and different interior models that require closure by the homogeneous wave equation in the far field. These experiments corroborate the theoretical design properties of the ABCs' algorithm.

## 2. Lacunae and Non-Deteriorating Numerical Integration.

**2.1. Lacunae of the Wave Equation.** We consider a Cauchy (initial-value) problem for the three-dimensional wave equation,  $\mathbf{x} = (x_1, x_2, x_3)$ :

$$\frac{\partial^2 \varphi}{\partial t^2} - c^2 \left( \frac{\partial^2 \varphi}{\partial x_1^2} + \frac{\partial^2 \varphi}{\partial x_2^2} + \frac{\partial^2 \varphi}{\partial x_3^2} \right) = f(\mathbf{x}, t), \quad t \geq 0, \quad (2.1)$$

$$\varphi \Big|_{t=0} = \frac{\partial \varphi}{\partial t} \Big|_{t=0} = 0. \quad (2.2)$$

(The limitation of having homogeneous initial conditions (2.2) can be alleviated, see [22, 23].) The problem (2.1), (2.2) is solved on the domain  $S(t) \subset \mathbb{R}^3$ , which has finite diameter  $d$  for all times  $t \geq 0$ ; other than that the domain  $S(t)$  may travel in space according to an arbitrary law of motion except that its maximum speed  $k$  is required to be “subsonic:”  $k < c$ . The solution  $\varphi(\mathbf{x}, t)$  is driven by the continuously operating source  $f(\mathbf{x}, t)$ ,  $f(\mathbf{x}, 0) = 0$ , and we require that  $\forall t > 0$ :  $\text{supp } f(\mathbf{x}, t) \subseteq S(t)$ . In other words, we study the radiation of waves by a source, which is compactly supported in space for all times. The solution is of interest for us also on a compact domain, which we call  $S(t)$ ; it fully contains the source and follows its motion if there is motion. This is a simplified model for many interesting physical phenomena that are more complex in their nature. As we shall see, this model is very useful when constructing the ABCs for a variety of problems.

For every  $(\mathbf{x}, t)$ , the solution  $\varphi = \varphi(\mathbf{x}, t)$  of problem (2.1), (2.2) is given by the Kirchhoff integral:

$$\varphi(\mathbf{x}, t) = \frac{1}{4\pi c^2} \iiint_{\varrho \leq ct} \frac{f(\boldsymbol{\xi}, t - \varrho/c)}{\varrho} d\boldsymbol{\xi}, \quad (2.3)$$

where  $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)$ ,  $\varrho = |\mathbf{x} - \boldsymbol{\xi}| = \sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2 + (x_3 - \xi_3)^2}$ , and  $d\boldsymbol{\xi} = d\xi_1 d\xi_2 d\xi_3$ . If we assume for a moment that the right-hand side (RHS)  $f(\mathbf{x}, t)$  is compactly supported in space-time on the domain  $Q \subset \mathbb{R}^3 \times [0, +\infty)$ , then formula (2.3) immediately implies that

$$\varphi(\mathbf{x}, t) \equiv 0 \quad \text{for } (\mathbf{x}, t) \in \bigcap_{(\boldsymbol{\xi}, \theta) \in Q} \{(\mathbf{x}, t) \mid |\mathbf{x} - \boldsymbol{\xi}| < c(t - \theta), t > \theta\}. \quad (2.4)$$

The region of space-time defined by formula (2.4) is called lacuna of the solution  $\varphi = \varphi(\mathbf{x}, t)$ . This region is obviously obtained as the intersection of characteristic cones of equation (2.1) once the vertex of the cone sweeps the support of the RHS:  $\text{supp } f \subseteq Q$ . From the standpoint of physics, the lacuna represents that part of space-time where the waves generated by the sources  $f$ ,  $\text{supp } f \subseteq Q$ , have already passed and the solution has become zero again. (Sometimes, the name “secondary lacuna” is used to distinguish it from the primary lacuna, which is the area where the waves have not reached yet.) The phenomenon of lacunae is inherently three-dimensional. The interior surface of the lacuna represents the trajectory of aft (trailing) fronts of the waves. The presence of aft fronts in odd-dimension spaces is known as the Huygens’ principle, as opposed to the so-called wave diffusion which takes place in even-dimension spaces.

**2.2. Computation with a Compactly Supported Source.** Assume now that the moment of inception of the source  $f(\mathbf{x}, t)$  is  $t_0$  (in particular it may be  $t_0 = 0$ ); at this moment the domain  $S(t_0)$  of the RHS  $f(\mathbf{x}, t)$  occupies a position in space which is schematically represented by the interval  $[A_1, A_2]$  of size  $d$  on Figure 2.1.<sup>2</sup> Assume also that by the time  $t_1 > t_0$  this source ceases to operate, which makes the RHS of

<sup>2</sup>Throughout this section we will be using schematic one-dimensional illustrations always keeping in mind, however, that the actual problem is three-dimensional.

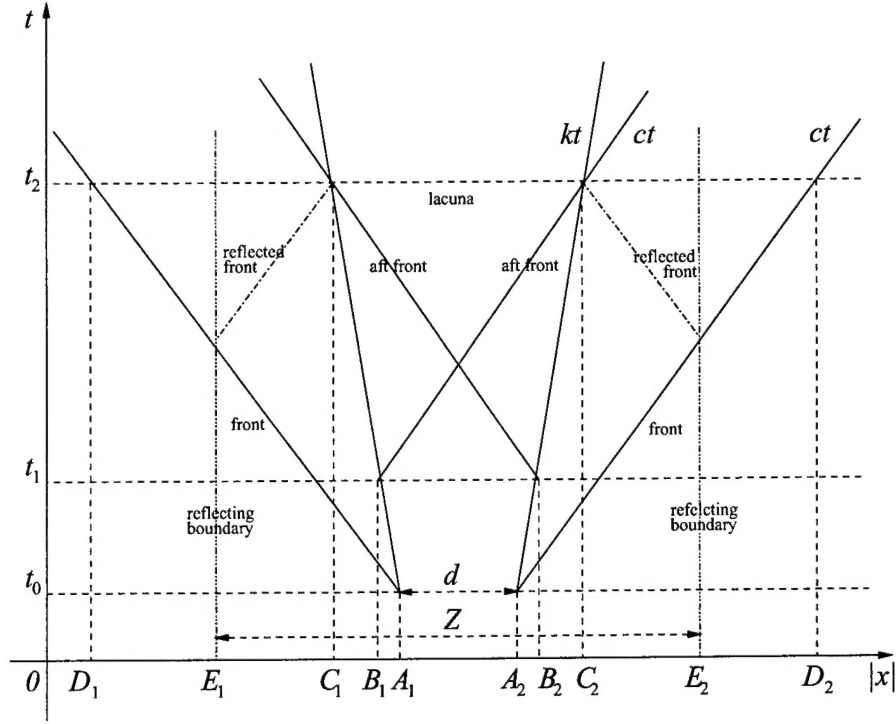


FIG. 2.1. 1D schematic representation for the fronts of waves generated by a compactly supported source.

equation (2.1) compactly supported in both space and time:  $\text{supp } f \subseteq Q = \{(x, t) | x \in S(t), t_0 < t < t_1\}$ . Clearly, by the time  $t_1$  the domain  $S(t_1)$  can be displaced from its initial location no further than the distance  $k(t_1 - t_0)$  in each direction, which is schematically represented on Figure 2.1 by the boundaries of the interval  $[B_1, B_2]$  of size  $d + 2k(t_1 - t_0)$ . Starting from  $t = t_1$  no new waves will be generated, and those generated prior to  $t_1$  will continue traveling in space and thus will eventually leave the domain  $S(t)$  completely, because their speed of propagation  $c$  is higher than that of the domain,  $k$ . The moment  $t_2$  when this happens, i.e., when the solution  $\varphi(x, t)$  again becomes zero on  $S(t)$ , is easy to calculate, see Figure 2.1. By this moment, the domain  $S(t_2)$  can travel no further than the interval  $[C_1, C_2]$  of size  $d + 2k(t_2 - t_0)$ , and we need to assume that the aft fronts will also be exactly at the boundaries of this interval at  $t = t_2$ , which immediately yields

$$\varphi(x, t) \equiv 0, \text{ for } x \in S(t), t \geq t_2 \equiv t_0 + \frac{d + (t_1 - t_0)(c + k)}{c - k}. \quad (2.5)$$

Estimate (2.5) is fundamental. It essentially says that once we need to calculate the solution  $\varphi(x, t)$  on  $S(t)$  and the sources are compactly supported in space-time:  $\text{supp } f \subseteq Q = \{(x, t) | x \in S(t), t_0 < t < t_1\}$ , then we may stop the calculation at  $t = t_2$  because afterwards the solution on  $S(t)$  will be zero anyway. This means, in particular, that if the solution is calculated using a discrete method, e.g., a finite-difference scheme, then no new error will be accumulated after  $t = t_2$ . The constants in both consistency and stability estimates of the scheme (see [22, 23] and below for detail) will now depend on the time interval  $T_{\text{int}} = \frac{d + (t_1 - t_0)(c + k)}{c - k}$  rather than final time  $T_{\text{final}}$ , which for the case of a compactly supported RHS simply becomes immaterial.

Besides, once we stop the calculation at  $t_2 = t_0 + T_{\text{int}}$ , we realize that during the time interval  $T_{\text{int}}$  that has passed since the beginning  $t = t_0$ , no waves could have traveled in space further than the boundaries

of the interval  $[D_1, D_2]$  of size  $d + 2cT_{\text{int}}$ , see Figure 2.1. Beyond this region the solution is zero because this is the area of the primary lacuna. Therefore, even though the original problem was formulated on an infinite domain, we can, in fact, calculate the solution on a finite domain  $[D_1, D_2]$  of size  $d + 2cT_{\text{int}}$  with zero external boundary conditions (of the Dirichlet type).

The transition from the infinite domain to a finite one does not, obviously, come “at no charge.” One can rather say that it comes at the expense of having the computational domain  $[D_1, D_2]$  larger than the actual domain of interest  $S(t)$ . However, the size of the “redundant” portion of  $[D_1, D_2]$  can be further reduced by observing that all we have to do is make sure that by  $t = t_2$ , i.e., by the moment the last waves generated by  $f(\mathbf{x}, t)$ ,  $\text{supp } f \subseteq Q$ , leave  $S(t)$ , no new waves can enter  $S(t)$ . This can be guaranteed either as indicated previously, by placing the outer boundary sufficiently far away so that no waves from  $f(\mathbf{x}, t)$ ,  $\text{supp } f \subseteq Q$ , can even reach it until  $t = t_2 = t_0 + T_{\text{int}}$  or alternatively, by placing it closer and thus allowing for reflections, but still not too close so that no reflected waves can come back, i.e., reach  $S(t)$ , by  $t = t_2$ . The size of the new, smaller, computational domain  $[E_1, E_2]$  with reflecting outer boundary, see Figure 2.1, can be estimated easily. The minimum size  $Z$ , see Figure 2.1, is found by requiring that the reflected waves, which travel with the same speed  $c$  but in the opposite direction, reach the boundary of  $[C_1, C_2]$ , i.e., the utmost possible location of  $S(t_2)$ , by the exact same moment of time  $t = t_2$  when the aft fronts leave  $S(t)$ . This immediately yields

$$Z = d + (k + c)T_{\text{int}} . \quad (2.6)$$

By comparing the value of  $Z$  from (2.6) with the size of  $[D_1, D_2]$ , which is  $d + 2cT_{\text{int}}$ , we conclude that the extra size of the computational domain beyond  $d$  can be reduced by up to a factor of 2 (when  $k = 0$ ) in each coordinate direction.

We also note that in fact any well-posed boundary condition can be specified at the reflecting outer boundary of  $[E_1, E_2]$ . The reason is that this boundary is intentionally positioned so that the reflections are not going to have any effect on the solution inside  $S(t)$  anyway. A particularly convenient way to treat the boundary of  $[E_1, E_2]$  will be to set the periodic boundary conditions there. In so doing the three-dimensional rectangular domain becomes a three-dimensional toroidal surface (the opposite faces of the rectangle are identified with one another) and we only have to keep in mind that the reflected waves will now need to be interpreted as those that leave the domain on one side and enter it from the opposite side. This new interpretation obviously brings no change to the foregoing considerations that led to the size estimate (2.6). However, for the case of a continuously operating traveling source that we analyze below, periodicity implies that the motion of the source can also be formally considered on the toroidal surface, which makes the computational setup much simpler.

**2.3. Computation with a Continuously Operating Source.** Both foregoing observations — finite time interval  $T_{\text{int}}$  and finite spatial domain  $[E_1, E_2]$  needed for calculating the solution driven by the sources  $f(\mathbf{x}, t)$ :  $\text{supp } f \subseteq Q = \{(\mathbf{x}, t) | \mathbf{x} \in S(t), t_0 < t < t_1\}$ , on the domain  $S(t)$  — are crucial for the original case of a continuously operating source  $f(\mathbf{x}, t)$ :  $\text{supp } f \subseteq \{(\mathbf{x}, t) | \mathbf{x} \in S(t), t \geq 0\}$ . In this case we first take a parameter  $T > 0$  and introduce a smooth even compactly supported function  $\Theta(t)$ ,  $t \in \mathbb{R}$ , of a “hat” type:

$$\begin{aligned} \Theta(t) &\equiv 0, \quad |t| \geq T, \\ \Theta(t) &= \Theta(-t), \\ \Theta(t) &\equiv 1, \quad t \in [-\sigma T, \sigma T], \quad 0 \leq \sigma < 1, \\ \Theta\left(\frac{1+\sigma}{2}T + t\right) &= 1 - \Theta\left(\frac{1+\sigma}{2}T - t\right), \end{aligned}$$



which obviously generates a partition of unity:

$$1 \equiv \sum_{j=0}^{\infty} \Theta(t - (1 + \sigma)Tj), \quad t \geq 0,$$

with the overlap size  $(1 - \sigma)T$ . Then, we represent the right-hand side  $f(\mathbf{x}, t)$  of equation (2.1) in the form

$$f(\mathbf{x}, t) = f(\mathbf{x}, t) \sum_{j=0}^{\infty} \Theta(t - (1 + \sigma)Tj) = \sum_{j=0}^{\infty} \Theta(t - (1 + \sigma)Tj) f(\mathbf{x}, t) = \sum_{j=0}^{\infty} f_j(\mathbf{x}, t, T). \quad (2.7)$$

Clearly, for each  $f_j(\mathbf{x}, t, T) \equiv \Theta(t - (1 + \sigma)Tj) f(\mathbf{x}, t)$ ,  $j = 0, 1, \dots$ , we have

$$\text{supp } f_j(\mathbf{x}, t, T) \subseteq \{(\mathbf{x}, t) | \mathbf{x} \in S(t), (j - 1 + \sigma j)T \leq t \leq (j + 1 + \sigma j)T\}. \quad (2.8)$$

Due to the linear superposition, the overall solution  $\varphi(\mathbf{x}, t)$  of equation (2.1) will be given by the sum of individual contributions from  $f_j(\mathbf{x}, t, T)$ ,  $j = 0, 1, \dots$ :

$$\varphi(\mathbf{x}, t) = \sum_{j=0}^{\infty} \varphi_j(\mathbf{x}, t, T), \quad (2.9)$$

where each contribution  $\varphi_j(\mathbf{x}, t, T)$  solves the following sub-problem:

$$\begin{aligned} \frac{\partial^2 \varphi_j}{\partial t^2} - c^2 \left( \frac{\partial^2 \varphi_j}{\partial x_1^2} + \frac{\partial^2 \varphi_j}{\partial x_2^2} + \frac{\partial^2 \varphi_j}{\partial x_3^2} \right) &= f_j(\mathbf{x}, t, T), \\ \varphi_j \Big|_{t=(j-1+\sigma j)T} &= \frac{\partial \varphi_j}{\partial t} \Big|_{t=(j-1+\sigma j)T} = 0, \quad j = 0, 1, 2, \dots \end{aligned} \quad (2.10)$$

Notice that each  $\varphi_j(\mathbf{x}, t, T)$ ,  $j = 0, 1, \dots$ , can be calculated absolutely independently of the others and that the corresponding source term  $f_j(\mathbf{x}, t, T)$  is a function compactly supported in both space and time, see (2.8). Consequently, according to (2.5), if we interpret  $t_0$  and  $t_1$  as  $(j - 1 + \sigma j)T$  and  $(j + 1 + \sigma j)T$ , respectively (see Figure 2.1), then we can conclude that every  $\varphi_j(\mathbf{x}, t, T)$  of (2.9) needs to be calculated only during a finite interval of time  $T_{\text{int}} = \frac{d+2T(c+k)}{c-k}$ . It is important to realize that this interval does not depend on the actual moment of time  $t$ .

Moreover, even so the series (2.9) is formally infinite, it is easy to see that for any  $t > 0$ ,  $\mathbf{x} \in S(t)$ , it contains only a finite fixed number of non-zero terms. First of all, because of the causality,  $\varphi_j(\mathbf{x}, t, T) = 0$  for  $\mathbf{x} \in S(t)$  if  $t < (j - 1 + \sigma j)T$ . In other words, for a given moment of time  $t$ , the contribution of all those  $f_j(\mathbf{x}, t, T)$  that are active only at subsequent moments of time, is obviously zero. A somewhat less trivial observation is that because of the lacunae the contribution of the "sufficiently retarded" terms  $f_j(\mathbf{x}, t, T)$  to the overall solution at a given time level  $t$  will be zero as well. More precisely,  $\varphi_j(\mathbf{x}, t, T) = 0$  for  $\mathbf{x} \in S(t)$  if  $(j - 1 + \sigma j)T < t - T_{\text{int}}$ . This follows immediately from (2.5) assuming that  $t_0 = (j - 1 + \sigma j)T$  and  $t_1 = (j + 1 + \sigma j)T$ . Consequently, instead of (2.9) we can write:

$$\varphi(\mathbf{x}, t) = \sum_{j=p_1}^{p_2} \varphi_j(\mathbf{x}, t, T), \quad \mathbf{x} \in S(t), \quad (2.11)$$

where  $p_1 = \left\lceil \frac{1}{1+\sigma} \left( \frac{t-T_{\text{int}}}{T} + 1 \right) \right\rceil$ ,  $p_2 = \left\lceil \frac{1}{1+\sigma} \left( \frac{t}{T} + 1 \right) \right\rceil$ , and  $\lceil \cdot \rceil$  stands for the integer part. The expressions for  $p_1$  and  $p_2$  indicate that we will always have either  $p_1 = p_2 - \left\lceil \frac{T_{\text{int}}}{(1+\sigma)T} \right\rceil$  or  $p_1 = p_2 - \left\lceil \frac{T_{\text{int}}}{(1+\sigma)T} \right\rceil - 1$ . Therefore, the number of terms  $p = p_2 - (p_1 - 1)$  in the sum (2.11) will never exceed  $\left\lceil \frac{T_{\text{int}}}{(1+\sigma)T} \right\rceil + 2$ . As  $T_{\text{int}}$  does not



depend on  $t$  we conclude that neither does the foregoing upper bound for  $p$ . As such, the number of terms in the sum (2.11) can always be considered finite and fixed. Altogether we obtain that  $\forall t > 0$  the solution  $\varphi(\mathbf{x}, t)$ ,  $\mathbf{x} \in S(t)$ , is composed of a finite non-increasing number  $p$  or additive terms, and each of the latter needs to be taken into account only during a finite non-increasing interval of time  $T_{\text{int}}$ .

In the perspective of numerical computation, the latter consideration translates into temporally uniform grid convergence of the discrete algorithm. Indeed, assume that we are integrating equation (2.1) by means of a finite-difference scheme with the order of accuracy  $\mathcal{O}(h^\alpha)$ , where  $h$  is a general notion for the grid size and  $\alpha > 0$ . Then, the discrete solution  $\varphi^{(h)}(\mathbf{x}, t)$  converges to the continuous solution  $\varphi(\mathbf{x}, t)$  as the grid size decreases:

$$\left\| \varphi^{(h)}(\mathbf{x}, t) - \varphi(\mathbf{x}, t) \right\| \leq K \cdot h^\alpha, \quad t \in [0, T_{\text{final}}], \quad (2.12)$$

here  $T_{\text{final}}$  is the total integration time. Inequality (2.12) is a generic convergence estimate; it holds provided that the RHS  $f(\mathbf{x}, t)$  of equation (2.1) is sufficiently smooth. A detailed discussion on the smoothness requirements for  $f(\mathbf{x}, t)$  can be found in [23], along with the specific consistency/stability/convergence estimates in the norms that would take into account a particular smoothness level.<sup>3</sup>

The constant  $K$  in inequality (2.12) does not depend on the grid. It is, however, known to depend on the actual RHS  $f(\mathbf{x}, t)$ , as well as the final time  $T_{\text{final}}$ :  $K = K(f, T_{\text{final}})$ . The dependency of  $K$  on  $T_{\text{final}}$  is typically a growth, and sometimes this growth may be rapid. This means that even so on any fixed interval  $[0, T_{\text{final}}]$  the scheme converges as  $h \rightarrow 0$ , to obtain the same level of accuracy on a larger  $[0, T_{\text{final}}]$  one may need to take a finer overall grid ahead of time. Thus, the convergence appears temporally non-uniform. On the language of practical computing, this phenomenon can be interpreted as the accumulation of numerical error over long runs. This issue has been long acknowledged unresolved in the literature.

The situation changes dramatically if, instead of the straightforward time-dependent integration, we first use the foregoing lacunae-based representation (2.11) of the solution  $\varphi(\mathbf{x}, t)$ . In so doing, for each  $j = p_1, \dots, p_2$ , we still integrate the corresponding sub-problem (2.10) using the same finite-difference scheme as before. However, convergence estimate for the scheme then becomes:

$$\left\| \varphi_j^{(h)}(\mathbf{x}, t, T) - \varphi_j(\mathbf{x}, t, T) \right\| \leq K_j \cdot h^\alpha, \quad (2.13)$$

$$\mathbf{x} \in S(t), \quad t \in [(j-1+\sigma j)T, (j-1+\sigma j)T + T_{\text{int}}].$$

A very important circumstance is that unlike  $K$  in estimate (2.12), the constant  $K_j$  in (2.13) for each  $j$  depends on  $T_{\text{int}}$  rather than  $T_{\text{final}}$ :  $K_j = K_j(f_j, T_{\text{int}})$ . Keeping in mind that each  $\varphi_j^{(h)}(\mathbf{x}, t, T)$  can be computed independently of the others, and using linear superposition (formula (2.10)), we then easily obtain instead of (2.12):

$$\left\| \varphi^{(h)}(\mathbf{x}, t) - \varphi(\mathbf{x}, t) \right\| \leq p \cdot \tilde{K} \cdot h^\alpha, \quad \mathbf{x} \in S(t), \quad t \geq 0, \quad (2.14)$$

where  $\tilde{K} = \tilde{K}(f, T_{\text{int}})$ . Note that  $p$  is fixed and does not increase with  $t$ , and  $\tilde{K}$  now depends on  $T_{\text{int}}$  rather than  $T_{\text{final}}$ , where  $T_{\text{int}}$  is also fixed and does not increase with  $t$ . Therefore, estimate (2.14) implies temporally uniform grid convergence of the discrete lacunae-based algorithm on arbitrarily long time intervals or in other words, for  $t \geq 0$ . A detailed formal proof of this result that, again, involves specific norms, can be found in [23]. Here we only need to add that for each inequality (2.13) to hold, the corresponding  $f_j(\mathbf{x}, t, T)$ ,

<sup>3</sup>Smoothness of the source terms will also be important when constructing the lacunae-based ABCs, see Sections 3 and 4.

$j = p_1, \dots, p_2$ , has to possess the same regularity as that required for the original scheme to converge. This explains why choosing the partition (2.7) smooth with overlaps was very important.

From the standpoint of practical computing, temporally uniform grid convergence implies that the numerical error will not get accumulated beyond some predetermined bound for as long as the computation needs to be performed; and once the grid is refined the aforementioned bound will also drop in accordance with the specific rate  $\mathcal{O}(h^\alpha)$ . This is clear because when the calculation is stopped for a given term  $\varphi_j^{(h)}(\mathbf{x}, t, T)$  after the interval  $T_{\text{int}}$  has elapsed, the error will not be accumulated any further, and the number of terms  $p$  that need to be taken into account is fixed and non-increasing. Thus, we have obtained a non-deteriorating numerical algorithm for integration of the wave equation over arbitrarily long times. Let us emphasize that it can be built as a modification of any consistent and stable finite-difference scheme, and that it preserves the original rate of convergence of the scheme while making the convergence uniform in time.

Besides, let us assume, for example, that the original finite-difference scheme has linear computational complexity with respect to the grid dimension, which is typical for explicit schemes. Then, it is easy to see that the modified lacunae-based algorithm will also have linear computational complexity with respect to the grid dimension. Indeed, this immediately follows from the fact that each term  $\varphi_j^{(h)}(\mathbf{x}, t, T)$  is computed using the original scheme on a compact domain of size  $Z$  (see Figure 2.1) during a finite fixed interval of time  $T_{\text{int}}$ , and the number of terms  $p$  is, again, fixed and non-increasing. We should note that for the type of problems that we are studying linear complexity with respect to the grid is, in fact, optimal, i.e., unimprovable.

**2.4. Computation Using Continuous Time Marching.** The following, and last, step in building the lacunae-based algorithm for long-term numerical integration of the wave equation is to realize that for implementing formula (2.11) we do not necessarily need to compute each term  $\varphi_j(\mathbf{x}, t, T)$  independently of the others. Instead, we can implement the algorithm in a way similar to the standard time-marching by means of a finite-difference scheme. For that, we will need to use the aforementioned periodic boundary conditions on the outer boundaries of the auxiliary domain  $[E_1, E_2]$ , see end of Section 2.2 and Figure 2.1.

The first key observation that we make here is that once the motion of the wave sources, as well as the propagation of waves themselves, are considered on a three-dimensional toroidal surface, rather than on the genuine  $\mathbb{R}^3$ , then for every portion of the RHS  $f_j(\mathbf{x}, t, T)$  it does not really matter where on the period this source is located, or where it starts its motion from, at  $t_0 = (j - 1 + \sigma j)T$ . It does not have to be exactly “in the middle” as shown on Figure 2.1, because all locations on the period (i.e., toroidal surface) are equivalent. All we have to worry about is that by the time  $t_2 = (j - 1 + \sigma j)T + T_{\text{int}}$  the waves generated by  $f_j(\mathbf{x}, t, T)$ , see (2.8), will have left the domain  $S(t)$ , and that no waves could have re-entered this domain during  $[t_0, t_2]$ . And this will be exactly the case because the size  $Z = d + (k + c)T_{\text{int}}$ , see (2.6), has been chosen sufficiently large to provide for that. Since we always assume (for simplicity) that the period  $Z$  is the same in all coordinate directions, then we only need to formally consider  $f_j(\tilde{\mathbf{x}}, t, T)$  instead of  $f_j(\mathbf{x}, t, T)$  and accordingly,  $\varphi_j(\tilde{\mathbf{x}}, t, T)$  instead of  $\varphi_j(\mathbf{x}, t, T)$ , where  $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$ , and  $\tilde{x}_i = x_i - \left\lfloor \frac{x_i}{Z} \right\rfloor Z$ ,  $i = 1, 2, 3$ .

Next, we shall analyze formula (2.11) from a slightly different point of view. To begin with, we notice that on the initial stage of computation, i.e., when  $t$  is small, the lower summation limit  $p_1$  may turn out negative. Basically, it does not create any inconsistency and does not cause any problem because  $f(\mathbf{x}, t) = 0$  for  $t < 0$  anyway. In fact, we can simply disregard all negative  $j$ 's in the sum (2.11) for small  $t$ 's and initially consider the summation  $\sum_{j=0}^{p_2} \varphi_j(\mathbf{x}, t, T)$  instead of (2.11). “Initially” here means till the actual expression  $p_1 = \left\lceil \frac{1}{1+\sigma} \left( \frac{t - T_{\text{int}}}{T} + 1 \right) \right\rceil$  becomes positive. It is easy to see that the computation on this initial stage is equivalent to the conventional time-marching of the wave equation (2.1) on the domain  $[E_1, E_2]$  of

size  $Z$  (see Figure 2.1) with periodic boundary conditions. Indeed, all we do here is simply take into account one component of the source  $f_j$  after another. Due to linear superposition, this amounts to the continuous integration of the wave equation driven by  $f = \sum_{j=0}^{p_2} f_j$  from  $t = 0$  till the actual time  $t$ . We also note that the duration of the initial stage is, obviously,  $T_{\text{int}}$ . And the period  $Z$ , see (2.6), has been chosen sufficiently large so that for the time interval of length  $T_{\text{int}}$  there will be no difference on the domain  $S(t)$  between the solution  $\varphi(\mathbf{x}, t)$  computed in the periodic setting and the solution that one could have possibly computed with no periodization (see Section 2.2).

As soon as the time interval  $T_{\text{int}} = \frac{d+2T(c+k)}{c-k}$  has elapsed since the inception moment  $t = 0$ , the computation enters its regular (as opposed to initial) stage. This regular stage, which can, in fact, be continued for as long as necessary, is characterized by the positive values of  $p_1$  (the first positive value is obviously  $p_1 = 1$ ) and finite non-growing number  $p = p_2 - (p_1 - 1)$  of terms in the sum (2.11).

On the regular stage of the algorithm, we continue marching equation (2.1) with periodic boundary conditions in space. Obviously, as the time  $t$  elapses both  $p_1$  and  $p_2$  in formula (2.11) increase. The increase of  $p_1$  and  $p_2$  is almost synchronous. Namely, as soon as  $t$  reaches the value  $(j - 1 + \sigma j)T$  for a particular integer  $j$ , a new term  $\varphi_j$  gets included into the sum (2.11), i.e., the upper summation bound  $p_2$  changes from its previous value  $j - 1$  to the new value  $j$ . Similarly, as soon as  $t$  reaches the value  $(j - 1 + \sigma j)T + T_{\text{int}}$  for a given  $j$ , the term  $\varphi_j$  drops from the sum (2.11), i.e., the lower summation bound changes from its previous value  $j$  to the new value  $j + 1$ . As has been mentioned in Section 2.3, in so doing the variation of the difference between  $p_2$  and  $p_1$  never exceeds one. Moreover, the temporal interval that precedes the actual moment  $t$  and that is taken into consideration by formula (2.11) is again  $T_{\text{int}}$ . Consequently, we can still compute everything in the periodic framework, because the period  $Z$  (see (2.6)) is sufficiently large to accommodate the extent of retardation  $T_{\text{int}}$ , and as has also been mentioned it does not matter where on the period the computation of every given term starts.

From the standpoint of implementation, when the upper bound  $p_2$  increases by one at  $t = (j - 1 + \sigma j)T$  nothing special needs to be done. If we simply continue marching equation (2.1) in the aforementioned periodic framework, then we will automatically start taking into account the new component of the RHS  $f_j$  after  $t = (j - 1 + \sigma j)T$ . The situation with the lower bound  $p_1$  is somewhat different. Once it has increased by one (from  $j$  to  $j + 1$ ) at  $t = (j - 1 + \sigma j)T + T_{\text{int}}$ , the term  $\varphi_j$  no longer needs to be included into the sum (2.11). However, in contradistinction to the case of Section 2.3 when all  $\varphi_j$  were supposed to be computed independently of one another, here we cannot just stop the computation of a given  $\varphi_j$  at  $t = (j - 1 + \sigma j)T + T_{\text{int}}$  and subsequently say that  $\varphi_j(\mathbf{x}, t, T) = 0$  for  $\mathbf{x} \in S(t)$  and for  $t \geq (j - 1 + \sigma j)T + T_{\text{int}}$ . Indeed, time-marching of equation (2.1) implies that all fragments of the solution  $\varphi_j(\mathbf{x}, t, T)$  are calculated together as a sum and cannot be explicitly told apart. On the other hand, if we do nothing at  $t = (j - 1 + \sigma j)T + T_{\text{int}}$  and continue with the time-marching, i.e., if we do not discontinue the computation of  $\varphi_j(\mathbf{x}, t, T)$  at  $t = (j - 1 + \sigma j)T + T_{\text{int}}$  and leave this term in the solution  $\varphi(\mathbf{x}, t)$ , then right after this moment of time the first waves generated by  $f_j$  at  $t = (j - 1 + \sigma j)T$  will start re-entering the domain  $S(t)$  having traveled all the way across the auxiliary domain  $[E_1, E_2]$ . In other words, in the framework of the continuous time-marching with periodic boundary conditions, the term  $\varphi_j(\mathbf{x}, t, T)$  cannot be left in the solution as it will “contaminate” the results on  $S(t)$ .

To avoid the aforementioned contamination, i.e., to prevent the re-entry of waves into  $S(t)$ , each term  $\varphi_j(\mathbf{x}, t, T)$  needs to be eliminated from the overall solution on the auxiliary domain  $[E_1, E_2]$  when the extent of its retardation (counted from inception) becomes exactly  $T_{\text{int}}$ . For a given  $j$  the proper moment of time for elimination of  $\varphi_j(\mathbf{x}, t, T)$  is  $t = (j - 1 + \sigma j)T + T_{\text{int}}$ . Once we take out  $\varphi_j(\mathbf{x}, t, T)$  at  $t = (j - 1 + \sigma j)T + T_{\text{int}}$ , this term may obviously be considered zero everywhere on  $[E_1, E_2]$  for all subsequent moments as well. To

take out the term  $\varphi_j(\mathbf{x}, t, T)$  we need to interrupt the time-marching at  $t = (j - 1 + \sigma j)T + T_{\text{int}}$ , then go back to the inception moment of  $f_j(\mathbf{x}, t, T)$ , which is  $t = (j - 1 + \sigma j)T$ , and independently integrate problem (2.10) for a particular  $j$  on  $[E_1, E_2]$  from  $t_0 = (j - 1 + \sigma j)T$  to  $t_2 = (j - 1 + \sigma j)T + T_{\text{int}}$ . The result should then be subtracted from the time-marching solution at  $t = t_2$  in the correct sense, i.e., both  $\varphi$  and  $\frac{\partial \varphi}{\partial t}$  (rather their discrete counterparts) should be affected. Alternatively, we may notice that when integrating problem (2.10) from  $t_0 = (j - 1 + \sigma j)T$  till  $t_2 = (j - 1 + \sigma j)T + T_{\text{int}}$ , the wave equation will, in fact, be homogeneous on a substantial portion of this time interval because  $f_j(\mathbf{x}, t, T) = 0$  for  $t > t_1 = (j + 1 + \sigma j)T$ . Consequently, instead of marching equation (2.10) over the entire time interval of length  $T_{\text{int}}$ , we may actually march it only from  $t_0 = (j - 1 + \sigma j)T$  till  $t_1 = (j + 1 + \sigma j)T$ , then Fourier transform the discrete solution and advance it till  $t_2 = (j - 1 + \sigma j)T + T_{\text{int}}$  by raising the corresponding amplification factors to the appropriate power. Numerically, this approach appears much cheaper, especially if it relies on highly efficient FFT subroutines.

The new version of the lacunae-based algorithm has obviously been designed so that to exactly reproduce the solution obtained with the original version of Section 2.3. The only difference is in the method of computation: Continuous time-marching in the periodic setup with cyclic subtractions of the retarded contributions versus separate computation of partial solutions driven by different components of the RHS. Consequently, the new version will possess the same properties as the original one. Foremost, it will provide for the temporally uniform grid convergence. Besides, it will obviously have linear computational complexity with respect to the grid dimension. (The cost of the FFT-based evolution in time distributed over the corresponding number of time steps is even less than linear if calculated per time step.) Finally, the algorithm will be universal in the sense that one will be able to build it as a modification of any consistent and stable finite-difference scheme. It will preserve the convergence rate of the original scheme while making the convergence uniform in time.

**2.5. Numerical Demonstrations.** To actually demonstrate that the lacunae-based algorithm is an appropriate procedure that does deliver according to its theoretical design properties, we present some numerical results for the wave equation. For our simulations, we assume axial symmetry and employ the  $(r, z)$  cylindrical coordinates so that to account for the three-dimensional effects using two-dimensional geometry. Accordingly, equation (2.1) becomes:

$$\frac{\partial^2 \varphi}{\partial t^2} - c^2 \left( \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial \varphi}{\partial r} \right) + \frac{\partial^2 \varphi}{\partial z^2} \right) = f(r, z, t), \quad t \geq 0. \quad (2.15)$$

The solution  $\varphi$  of equation (2.15), as well as the RHS  $f$ , are functions of  $r$ ,  $z$ , and  $t$ . The initial conditions for equation (2.15) remain homogeneous as before, see (2.2).

We introduce the rectangular auxiliary domain  $[0, R] \times [-Z/2, Z/2]$  of variables  $(r, z)$ , this domain is a specific realization of  $[E_1, E_2]$  shown in Figure 2.1. The boundary conditions are periodic with the period  $Z$  in the  $z$  direction, and zero Dirichlet at  $r = R$ :

$$\begin{aligned} \varphi(r, z \pm Z, t) &= \varphi(r, z, t), \\ \varphi(R, z, t) &= 0. \end{aligned} \quad (2.16)$$

The mathematical formulation of the problem obviously requires no boundary conditions at  $r = 0$ . However, for the purpose of subsequently building a discrete scheme (see below) we notice that the natural assumption of  $\varphi(r, z, t)$  being a bounded smooth function, along with the axial symmetry, immediately imply that  $\frac{\partial \varphi}{\partial r} \Big|_{r=0} = 0$ . Consequently, the Taylor expansion for  $\varphi$  near  $r = 0$  yields:

$$\varphi(r, \cdot) = \varphi(0, \cdot) + \frac{1}{2} \frac{\partial^2 \varphi}{\partial r^2} \Big|_{r=0} \cdot r^2 + \mathcal{O}(r^3),$$

which means that

$$\frac{\partial \varphi}{\partial r} = \frac{\partial^2 \varphi}{\partial r^2} \Big|_{r=0} \cdot r + \mathcal{O}(r^2) .$$

Substituting the latter expression into (2.15) and considering the limit  $r \rightarrow 0$ , we obtain that on the  $z$ -axis, i.e., at  $r = 0$ , equation (2.15) reduces to:

$$\frac{\partial^2 \varphi}{\partial t^2} - c^2 \left( 2 \frac{\partial^2 \varphi}{\partial r^2} + \frac{\partial^2 \varphi}{\partial z^2} \right) = f(r, z, t) , \quad r = 0 , \quad t \geq 0 . \quad (2.17)$$

To assess the quality of our numerical method we need to build a reference exact solution of problem (2.15), (2.2). This solution is obtained using the Lorentz' transform:

$$\begin{aligned} \theta &= \frac{1}{\sqrt{1 - k^2/c^2}} \cdot t - \frac{k/c}{\sqrt{1 - k^2/c^2}} \cdot \frac{z}{c} , \\ \zeta &= -\frac{k/c}{\sqrt{1 - k^2/c^2}} \cdot ct + \frac{1}{\sqrt{1 - k^2/c^2}} \cdot z . \end{aligned} \quad (2.18)$$

Transformation (2.18) introduces the new coordinate system  $(r, \zeta, \theta)$ . The origin of this new coordinate system moves with the speed  $k$  along the  $z$ -axis of the original coordinate system. In other words, at every given  $t$  it is positioned at  $z = kt$  in the original frame of reference. In implementing transformation (2.18), we will always need to assume that  $k < c$ , as has also been suggested in Section 2.2.

The key property of the Lorentz' transform (2.18) is that it does not change the form of the wave equation (2.1) (and consequently, (2.15) and (2.17)), see, e.g., [25]. As such, let us introduce an arbitrary function of time  $\chi = \chi(t)$ ,  $\chi(t) = 0$  for  $t \leq 0$ , so that it also be smooth on the entire  $\mathbb{R}$ . Next, we define  $\rho^2 = r^2 + \zeta^2$ , and then

$$\psi(r, \zeta, \theta) = \frac{\chi\left(\theta - \frac{\rho}{c}\right)}{\rho} \quad (2.19a)$$

becomes a solution to the wave equation in the new coordinates  $(r, \zeta, \theta)$ . Solution (2.19a) is driven by a point  $\delta$ -type source, which is located at the origin  $\{r = 0, \zeta = 0\}$  and modulated in time by the function  $\chi(\theta)$ . As  $\chi'(0) = 0$ , this solution also satisfies the homogeneous initial conditions. Consequently, the function

$$\psi(r, z, t) = \frac{\chi\left(\theta(z, t) - \frac{\rho(r, z, t)}{c}\right)}{\rho(r, z, t)} \quad (2.19b)$$

obtained by substituting (2.18) into (2.19a) is a solution of equation (2.15) with the RHS  $f(r, z, t) = \chi(t) \cdot \delta(r, z - kt)$ . In other words,  $\psi(r, z, t)$  of (2.19b) is a solution to the wave equation excited by a  $\delta$ -source that moves straightforwardly and uniformly and is modulated in time by a given smooth function. Solution (2.19b) also satisfies homogeneous initial conditions (2.2). From the standpoint of physics, solution (2.19b) can be characterized as radiation of spherical waves by a moving point source.

Solution (2.19b) is obviously singular. To use it for testing the numerical algorithm we need to remove the singularity. For that, let us first introduce the actual domain  $S(t)$ . In all the experiments that follow, the domain  $S(t)$  is a sphere of diameter  $d$  with its center at the origin of the new coordinate system:  $\{r = 0, z = kt\}$ . As such, this spherical domain moves uniformly along the  $z$ -axis, which obviously helps us keep the axial symmetry intact. As has been mentioned, the speed of this motion is "subsonic,"  $k < c$ , which conforms to one of the key requirements for building the lacunae-based algorithm previously put forward in Section 2.2.

Let us also define  $\tilde{r}^2 = r^2 + (z - kt)^2$  and introduce the function  $Q = Q(\tilde{r})$ ,  $\tilde{r} \geq 0$ , such that  $Q(0) = 0$ ,  $Q(\tilde{r}) \equiv 1$  for  $\tilde{r} \geq \kappa d/2$ , where  $\kappa < 1$ , and also  $\frac{d^m Q(0)}{d\tilde{r}^m} = \frac{d^m Q(\kappa d/2)}{d\tilde{r}^m} = 0$  for  $m = 1, 2, \dots$  till at least  $m = 4$ . Then, it is easy to see that the function  $\varphi(r, z, t) = \psi(r, z, t) \cdot Q(\tilde{r})$  is regular (continuous and bounded) everywhere. Moreover, it is easy to verify by direct differentiation that the same is true for the function  $f(r, z, t) \stackrel{\text{def}}{=} \square \varphi(r, z, t)$ , where  $\square$  denotes the wave operator, i.e., the left-hand side of equation (2.15). We will use  $f(r, z, t)$  defined this way as the source function for equation (2.15). Clearly,  $f(r, z, t)$  may, generally speaking, differ from zero only on the ball of a smaller diameter  $\kappa d$  concentric with  $S(t)$ . Everywhere else, i.e., for  $\tilde{r} > \kappa d/2$ ,  $f(r, z, t) = 0$ .

Obviously, the solution of problem (2.15), (2.2) driven by this  $f(r, z, t) = \square \varphi(r, z, t)$  is the foregoing

$$\varphi(r, z, t) = \psi(r, z, t) \cdot Q(\tilde{r}). \quad (2.20)$$

This function satisfies the non-homogeneous wave equation with the RHS  $f(r, z, t)$  on a smaller ball of diameter  $\kappa d$  concentric with  $S(t)$ . Everywhere else it is a solution to the homogeneous wave equation because it coincides with  $\psi(r, z, t)$  of (2.19b). Consequently,  $\varphi(r, z, t)$  of (2.20) can be interpreted as the radiation of waves by a compactly supported moving source  $f(r, z, t)$ . Numerically, we will be reproducing solution  $\varphi(r, z, t)$  given by (2.20) on the domain  $S(t)$  using finite-difference methods.

We employ three different explicit central-difference schemes in our simulations. In all three cases we construct a uniform rectangular grid on the plane  $(r, z)$ :  $r_l = lh_r$ ,  $l = 0, 1, \dots, N_r$ ,  $h_r = R/N_r$ , and  $z_m = mh_z$ ,  $m = 0, \pm 1, \dots, \pm N_z$ ,  $h_z = Z/2N_z$ . The discrete time levels are  $t_n = n\tau$ ,  $n = 0, 1, \dots$ . For the cell-centered second-order scheme, we keep the values of the unknown function  $\varphi$  at the grid nodes in the  $z$  direction and at mid-points in the  $r$  direction:

$$\begin{aligned} & \frac{\varphi_{l+1/2,m}^{n+1} - 2\varphi_{l+1/2,m}^n + \varphi_{l+1/2,m}^{n-1}}{\tau^2} - \\ & c^2 \left( \frac{1}{r_{l+1/2}} \frac{1}{h_r} \left[ r_{l+1} \frac{\varphi_{l+3/2,m}^n - \varphi_{l+1/2,m}^n}{h_r} - r_l \frac{\varphi_{l+1/2,m}^n - \varphi_{l-1/2,m}^n}{h_r} \right] + \right. \\ & \left. \frac{\varphi_{l+1/2,m+1}^n - 2\varphi_{l+1/2,m}^n + \varphi_{l+1/2,m-1}^n}{h_z^2} \right) = f_{l+1/2,m}^n. \end{aligned} \quad (2.21a)$$

Equations (2.21a) hold for all  $l > 0$ . As in this case we do not have the unknown function defined on the axis of symmetry, and the closest values that correspond to  $l = 0$  are half-grid-size away:  $\varphi_{1/2,m}^n$ , then the scheme for  $l = 0$  is obtained by simply assuming that  $\frac{\varphi_{l+1/2,m}^n - \varphi_{l-1/2,m}^n}{h_r} \Big|_{l=0} = 0$ , which can be interpreted as a second-order approximation of the natural condition  $\frac{\partial \varphi}{\partial r} \Big|_{r=0} = 0$ . This immediately yields for  $l = 0$ :

$$\begin{aligned} & \frac{\varphi_{1/2,m}^{n+1} - 2\varphi_{1/2,m}^n + \varphi_{1/2,m}^{n-1}}{\tau^2} - \\ & c^2 \left( \frac{1}{r_{1/2}} \frac{1}{h_r} r_1 \frac{\varphi_{3/2,m}^n - \varphi_{1/2,m}^n}{h_r} + \frac{\varphi_{1/2,m+1}^n - 2\varphi_{1/2,m}^n + \varphi_{1/2,m-1}^n}{h_z^2} \right) = f_{1/2,m}^n. \end{aligned} \quad (2.21b)$$

For the node-centered second-order scheme,  $\varphi$  is taken at the actual grid nodes, and for  $l > 0$  we have:

$$\begin{aligned} & \frac{\varphi_{l,m}^{n+1} - 2\varphi_{l,m}^n + \varphi_{l,m}^{n-1}}{\tau^2} - \\ & c^2 \left( \frac{1}{r_l} \frac{1}{h_r} \left[ r_{l+1/2} \frac{\varphi_{l+1,m}^n - \varphi_{l,m}^n}{h_r} - r_{l-1/2} \frac{\varphi_{l,m}^n - \varphi_{l-1,m}^n}{h_r} \right] + \right. \\ & \left. \frac{\varphi_{l,m+1}^n - 2\varphi_{l,m}^n + \varphi_{l,m-1}^n}{h_z^2} \right) = f_{l,m}^n. \end{aligned} \quad (2.22a)$$

To obtain the scheme on the axis of symmetry  $l = 0$  in this case, we need to approximate equation (2.17). For the  $\frac{\partial^2}{\partial r^2}$  derivative in this equation we can first formally write  $\frac{\partial^2 \varphi}{\partial r^2} \Big|_{r=0} \approx \frac{\varphi_{1,m} - 2\varphi_{0,m} + \varphi_{-1,m}}{h_r^2}$ . This expression obviously reduces to  $\frac{\partial^2 \varphi}{\partial r^2} \Big|_{r=0} \approx 2 \frac{\varphi_{1,m} - \varphi_{0,m}}{h_r^2}$  because of the symmetry:  $\varphi_{-1,m} = \varphi_{1,m}$ , and consequently, we obtain:

$$\begin{aligned} & \frac{\varphi_{l,m}^{n+1} - 2\varphi_{l,m}^n + \varphi_{l,m}^{n-1}}{\tau^2} - \\ & c^2 \left( 4 \frac{\varphi_{1,m}^n - \varphi_{0,m}^n}{h_r^2} + \frac{\varphi_{0,m+1}^n - 2\varphi_{0,m}^n + \varphi_{0,m-1}^n}{h_z^2} \right) = f_{0,m}^n. \end{aligned} \quad (2.22b)$$

The last scheme is the node-centered fourth-order scheme. More precisely, it approximates spatial derivatives with the accuracy  $\mathcal{O}(h_r^4 + h_z^4)$  and temporal derivative with the accuracy  $\mathcal{O}(\tau^2)$ . For  $l > 1$  we have:

$$\begin{aligned} & \frac{\varphi_{l,m}^{n+1} - 2\varphi_{l,m}^n + \varphi_{l,m}^{n-1}}{\tau^2} - \\ & c^2 \left( \frac{4}{3} \frac{1}{r_l} \frac{1}{h_r} \left[ r_{l+1/2} \frac{\varphi_{l+1,m}^n - \varphi_{l,m}^n}{h_r} - r_{l-1/2} \frac{\varphi_{l,m}^n - \varphi_{l-1,m}^n}{h_r} \right] - \right. \\ & \quad \frac{1}{3} \frac{1}{r_l} \frac{1}{2h_r} \left[ r_{l+1} \frac{\varphi_{l+2,m}^n - \varphi_{l,m}^n}{2h_r} - r_{l-1} \frac{\varphi_{l,m}^n - \varphi_{l-2,m}^n}{2h_r} \right] + \\ & \quad \left. \frac{-\varphi_{l,m+2}^n + 16\varphi_{l,m+1}^n - 30\varphi_{l,m}^n + 16\varphi_{l,m-1}^n - \varphi_{l,m-2}^n}{12h_z^2} \right) = f_{l,m}^n. \end{aligned} \quad (2.23a)$$

For  $l = 1$  we have  $r_{l-1} = r_0 = 0$  and consequently:

$$\begin{aligned} & \frac{\varphi_{1,m}^{n+1} - 2\varphi_{1,m}^n + \varphi_{1,m}^{n-1}}{\tau^2} - \\ & c^2 \left( \frac{4}{3} \frac{1}{r_1} \frac{1}{h_r} \left[ r_{3/2} \frac{\varphi_{2,m}^n - \varphi_{1,m}^n}{h_r} - r_{1/2} \frac{\varphi_{1,m}^n - \varphi_{0,m}^n}{h_r} \right] - \frac{1}{3} \frac{1}{r_1} \frac{1}{2h_r} \left[ r_2 \frac{\varphi_{3,m}^n - \varphi_{1,m}^n}{2h_r} \right] + \right. \\ & \quad \left. \frac{-\varphi_{1,m+2}^n + 16\varphi_{1,m+1}^n - 30\varphi_{1,m}^n + 16\varphi_{1,m-1}^n - \varphi_{1,m-2}^n}{12h_z^2} \right) = f_{1,m}^n. \end{aligned} \quad (2.23b)$$

Finally, for  $l = 0$  we again have to approximate equation (2.17). Using symmetry like in the previous case, we arrive at:

$$\begin{aligned} & \frac{\varphi_{0,m}^{n+1} - 2\varphi_{0,m}^n + \varphi_{0,m}^{n-1}}{\tau^2} - \\ & c^2 \left( 2 \frac{-2\varphi_{2,m}^n + 32\varphi_{1,m}^n - 30\varphi_{0,m}^n}{12h_r^2} + \right. \\ & \quad \left. \frac{-\varphi_{0,m+2}^n + 16\varphi_{0,m+1}^n - 30\varphi_{0,m}^n + 16\varphi_{0,m-1}^n - \varphi_{0,m-2}^n}{12h_z^2} \right) = f_{0,m}^n. \end{aligned} \quad (2.23c)$$

For all three schemes, (2.21), (2.22), and (2.23), setting the discrete boundary conditions (2.16) on the outer boundary of the auxiliary domain  $[0, R] \times [-Z/2, Z/2]$  is straightforward. An extra boundary condition is needed for the fourth-order approximation. As it basically does not matter what boundary conditions we use on the outer boundary of the auxiliary domain (see Section 2.2), we simply set  $\varphi_{N_r-1,m}^n = 0$  in addition to  $\varphi_{N_r,m}^n = 0$ . Regarding the time step  $\tau$ , all three schemes are explicit and as such, there is a Courant-type stability constraint.

As has been mentioned, we present the results of numerical computations that follow in order to corroborate the theoretical design properties of the lacunae-based algorithm, i.e., to show the temporally uniform



grid convergence on long time intervals. For that purpose, we conduct a grid refinement study, i.e., approximate the exact solution (2.20) on a sequence of successively more fine grids. In so doing, the time step  $\tau$  for the two second-order schemes (2.21) and (2.22) is always reduced with the same rate as the corresponding spatial sizes  $h_r$  and  $h_z$ ; and the time step  $\tau$  for the fourth-order scheme (2.23) is reduced twice as fast (i.e., by a factor of four every time  $h_r$  and  $h_z$  are reduced by a factor of two) so that to demonstrate the fourth-order overall convergence in the end. The computations in each case were run till the dimensionless time  $t = 200 \cdot d/c$ , i.e., for 200 times the time interval required for a wave to cross the domain. This certainly qualifies as “long term” from the standpoint of any conceivable application.

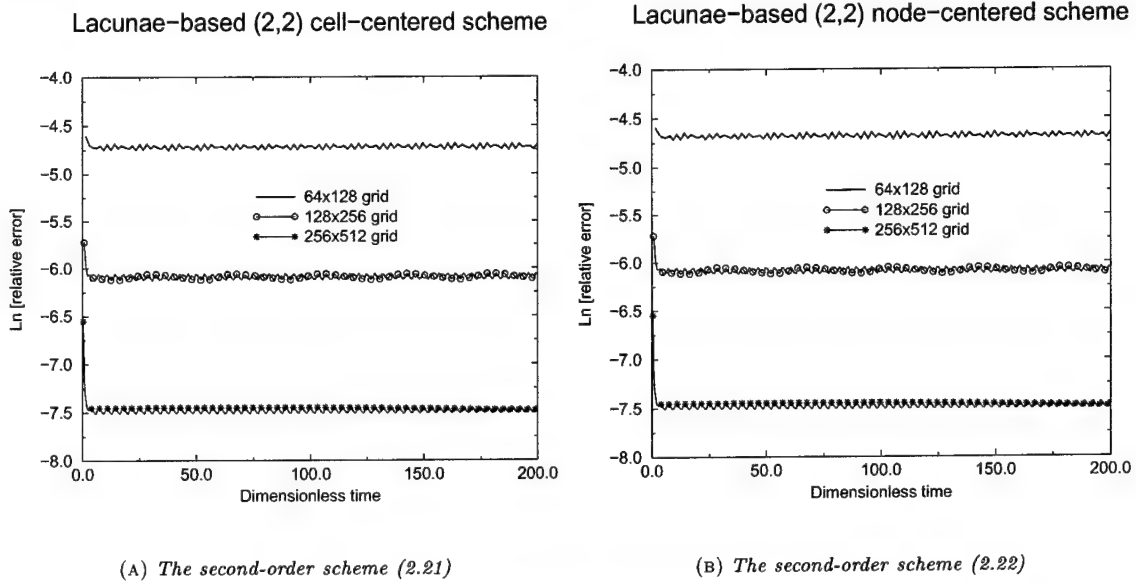


FIG. 2.2. Grid convergence study for the long-term lacunae-based integration of the wave equation.

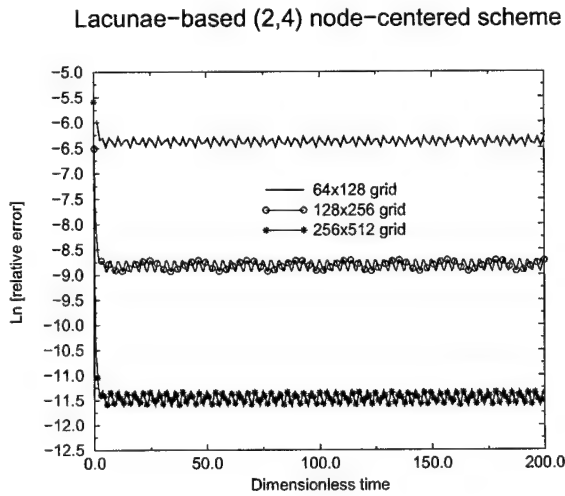


FIG. 2.3. Same as Fig. 2.2 for the fourth-order scheme (2.23).

The actual parameters that we have used for our simulations are the following:  $R = Z = \pi$ ,  $d = 1.8$ ,  $c = 1$ ,  $k = 0.2$ ,  $\kappa = 0.8$ . The spatial grid is composed of square cells:  $N_r = N_z$  and consequently,  $h_r = h_z = h$ . The actual grid dimensions  $N_r \times 2N_z$  are:  $64 \times 128$ ,  $128 \times 256$ , and  $256 \times 512$ . The temporal partition size  $2T$ , see (2.8), is found from (2.6) assuming that  $T_{\text{int}} = \frac{d+2T(c+k)}{c-k}$ ; the overlap parameter  $\sigma = 1/2$ . The functions  $\Theta(t)$  and  $Q(\vec{r})$  on the intervals of their variation from 0 to 1 are built as polynomials of degree 9 (with only odd powers included), which guarantees four continuous derivatives in transition to the constant (either 0 or 1). The function  $\chi(t)$  is defined as follows:  $\chi(t) = (1 + \frac{1}{4} \sin t) P(1 - \frac{t}{2\pi})$ , where  $P(t)$  is, again, a polynomial of degree 9 that decays

smoothly from 1 to 0 on the interval  $[0, 1]$  (four continuous derivatives). Finally, to subtract every  $\varphi_j$  from the overall solution at the proper moment  $t = (j - 1 + \sigma j)T + T_{\text{int}}$ , we first march equation (2.10) from  $t = (j - 1 + \sigma j)T$  till  $t = (j + 1 + \sigma j)T$  and then use Fourier expansion in  $z$  and expansion with respect to the corresponding discrete eigenfunctions (calculated numerically) in  $r$  to advance it further till  $t = (j - 1 + \sigma j)T + T_{\text{int}}$ .

In Figure 2.2 we show error profiles (more precisely, natural logarithm of the relative error on the domain  $S(t)$  in the maximum norm as it depends on the dimensionless time) on all three grids for both second-order schemes (2.21) and (2.22). In Figure 2.3, similar curves are shown for the fourth-order scheme (2.23). From these figures we conclude that indeed no error is accumulated in the course of computations because all error profiles are flat throughout the entire  $200 \cdot d/c$  time interval. Thus, the solution does not deteriorate as time elapses. Figure 2.2 also shows that every time the grid is refined by a factor of two the error drops by approximately a factor of four, which indicates the second-order convergence. Similarly, Figure 2.3 shows that every time the grid is refined by a factor of two the error drops by approximately a factor of sixteen, which is an indication of the fourth-order convergence. Consequently, we can conclude that numerical experiments fully corroborate the theoretical design properties of the algorithm.

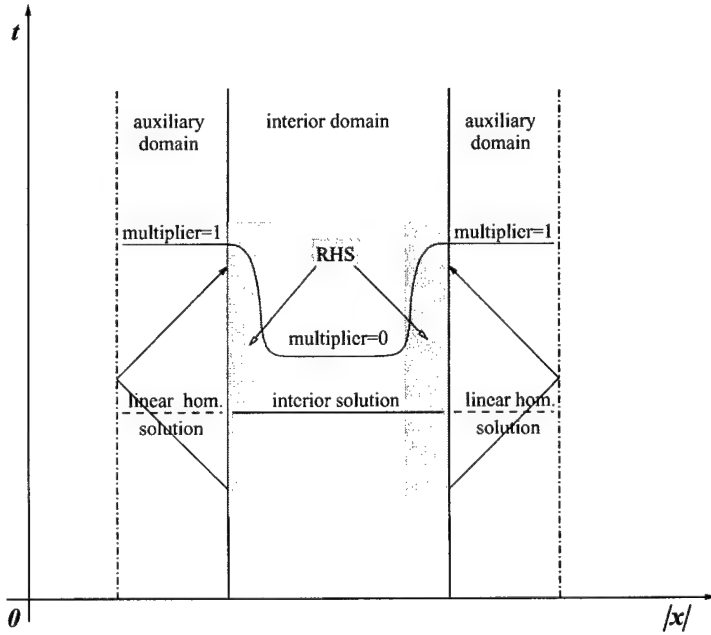


FIG. 3.1. Schematic geometric setup for the ABCs.

### 3. Lacunae-Based ABCs for the Wave Equation.

The lacunae-based algorithm of Section 2 provides a venue for constructing the ABCs for a class of problems that reduce to the homogeneous wave equation in the far field. We schematically depict the geometric setup for one such problem in Figure 3.1, assuming for simplicity that there is no source motion,  $k = 0$ , and the computational domain is stationary. We emphasize though that this is not a limitation, and that the actual ABCs will be constructed and tested for the general case of a moving computational domain, while the law of motion can be arbitrary, see Section 2.1. The problem to be solved on the bounded interior domain, i.e., in the near field, see Figure 3.1, may involve some complex

phenomena whose nature, however, is not essential for the current discussion.<sup>4</sup> We only require that the overall combined formulation of the problem be uniquely solvable and well posed under the assumption of radiation of waves in the far field (from  $S(t)$  toward infinity), where the problem is assumed to be governed by the homogeneous wave equation. The role of the ABCs (as mentioned in Section 1) is to provide a closure for the truncated problem solved on the actual computational domain  $S(t)$ . This closure has to ensure that

<sup>4</sup>The interior domain is, of course, the same as  $S(t)$  of Section 2; for the stationary case we obviously have  $S(t) \equiv S(0)$ .

the corresponding finite-domain solution recovered with the help of the ABCs be close to (ideally, exactly the same as) the solution of the original non-truncated problem restricted to the bounded domain, see [2].

In the wave propagation framework adopted in this paper, one can say that the ABCs have to replace the entire far field, i.e., everything beyond the bounded interior domain  $S(t)$ , so that the resulting artificial boundary be completely transparent for all the outgoing (i.e., radiated) waves. We also note that the incoming waves, provided that they are meaningful for a particular setup, can, in fact, be taken into account through the boundary conditions as well, but we do not discuss this issue here for the reason of simplicity.

**3.1. Preliminary Considerations in the Continuous Framework.** Let  $\varphi_c = \varphi_c(\mathbf{x}, t)$  be a solution to the aforementioned combined problem. In the far field, i.e., outside  $S(t)$ , the function  $\varphi_c(\mathbf{x}, t)$  satisfies the homogeneous wave equation. We also assume for simplicity that the solution  $\varphi_c(\mathbf{x}, t)$  “smoothly originates from zero” at  $t = 0$ , (i.e., turns into zero along with its first derivative) in much the same way as the solution  $\varphi(\mathbf{x}, t)$  of (2.1), (2.2) does. This assumption, in fact, will present no limitation when constructing the ABCs. The argument is the same as the one that allows us to relax the assumption of homogeneity of initial conditions when building the original lacunae-based algorithm, see [23].

Let us now introduce a special multiplier function that is again schematically shown in Figure 3.1. This function  $\mu = \mu(\mathbf{x}, t)$  is defined for all those  $\mathbf{x}$  and  $t$ , for which the solution  $\varphi_c(\mathbf{x}, t)$  makes sense. We first require that  $\forall t > 0, \forall \mathbf{x} \notin S(t) : \mu(\mathbf{x}, t) \equiv 1$ , or in other words, that the multiplier be identically equal to one everywhere outside the computational domain  $S(t)$  for all times. We also require that the multiplier be identically equal to zero,  $\mu(\mathbf{x}, t) \equiv 0$ , on most of the domain  $S(t)$  (again, for every  $t$ ) except next to its boundary from the interior side. An example of the narrow near-boundary transition region, where the multiplier  $\mu(\mathbf{x}, t)$  changes its value from zero to one, is shaded on Figure 3.1. What is important, we require that the multiplier  $\mu(\mathbf{x}, t)$  be a sufficiently smooth function with respect to both  $\mathbf{x}$  and  $t$ , which essentially means that the transition within the shaded region on Figure 3.1 has to be smooth. Regarding the time dependency of  $\mu(\mathbf{x}, t)$ , once the domain  $S(t)$  moves according to a prescribed law, the construction of the multiplier has to trace that motion. If the computational domain is stationary,  $S(t) \equiv S(0)$ , then the multiplier still may, but does not have to, depend on time.

Next, we apply the wave operator  $\square \equiv \frac{\partial^2}{\partial t^2} - c^2 \Delta$  of (2.1) to the function  $\mu(\mathbf{x}, t) \cdot \varphi_c(\mathbf{x}, t)$ , which is defined everywhere, i.e., both inside and outside  $S(t)$ .<sup>5</sup> We will obviously have:

$$\square(\mu\varphi_c) = \frac{\partial^2 \mu\varphi_c}{\partial t^2} - c^2 \Delta(\mu\varphi_c) = g(\mathbf{x}, t) \begin{cases} = 0 & \forall t, \forall \mathbf{x} \notin S(t) \\ \neq 0 & \text{in the transition region} \\ = 0 & \text{“well inside” } S(t) \end{cases} \quad (3.1)$$

The function  $g(\mathbf{x}, t)$  of (3.1) may generally speaking differ from zero only in the foregoing near-boundary transition region; it is zero outside  $S(t)$  because the function  $\mu\varphi_c$  coincides there with the solution  $\varphi_c$  of the homogeneous wave equation; it is also zero inside  $S(t)$  because  $\mu = 0$  there. On Figure 3.1 the non-zero portion of  $g(\mathbf{x}, t)$  is identified as the right-hand side RHS.

We can now consider the problem (2.1), (2.2) with the function  $g(\mathbf{x}, t)$  of (3.1) substituted instead of the generic RHS  $f(\mathbf{x}, t)$ . The key fact that we will need for constructing the ABCs, and that follows immediately from the unique solvability of the Cauchy problem for the wave equation, is that the solution to this problem will coincide with  $\mu(\mathbf{x}, t) \cdot \varphi_c(\mathbf{x}, t)$  everywhere. What will be of particular importance to us is that as such,

<sup>5</sup>Note, the solution  $\varphi_c(\mathbf{x}, t)$  may not be defined on all of  $S(t)$  if, e.g., there is a scatterer inside. As, however,  $\mu(\mathbf{x}, t) = 0$  there, we can consider  $\mu(\mathbf{x}, t) \cdot \varphi_c(\mathbf{x}, t)$  to be defined everywhere.

this solution will coincide with  $\varphi_c(\mathbf{x}, t)$  outside  $S(t)$  for all times, because  $\mu(\mathbf{x}, t) \equiv 1$  there. In other words, we have replaced all of the interior problem on  $S(t)$  (no matter how complex it may be) by the special near-boundary source function  $g(\mathbf{x}, t)$  so that the corresponding far-field portion of the solution, i.e., the wave-radiation solution outside  $S(t)$ , remains totally unaffected. Later on, see Sections 3.2 and 3.3, this reduction interpreted in the discrete framework will be used for setting the ABCs. The idea is to use the exterior solution obtained in an alternative way through integrating the near-boundary sources as a closure for the interior problem solved on the finite computational domain.

**3.2. The Concept of Discrete ABCs.** To construct the ABCs for a finite-difference scheme that approximates the problem described in the beginning of Section 3, we will employ the considerations similar to those of Section 3.1, but on the discrete level. As a helpful illustration, we will first consider here a one-dimensional model example,<sup>6</sup> and then, in the next Section 3.3, show how to build the ABCs for the actual multidimensional wave propagation problems.

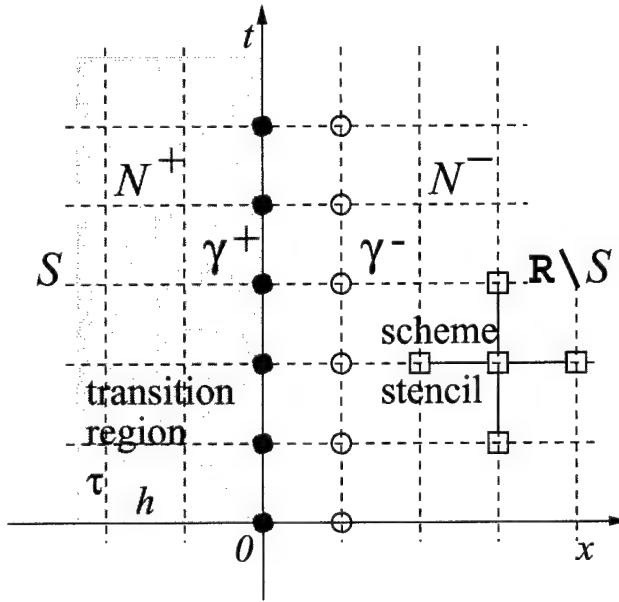


FIG. 3.2. Illustration to the one-dimensional example.

Assume that we are solving a one-dimensional combined problem on the entire  $\mathbb{R}$ . The computational domain  $S(t) \equiv S$  (i.e., the “near field”) is fixed, it is the half-line  $x \leq 0$  (more precisely, it is  $\{(x, t) | x \leq 0, t \geq 0\}$ ); its complement  $\mathbb{R} \setminus S$  represents the “far field,” which is to be truncated and replaced with the ABCs. As such, the ABCs are to be set at the interface  $x = 0$ . In accordance with the previous discussion, we also assume that the far field is governed by the one-dimensional homogeneous wave equation

$$\frac{\partial^2 \varphi}{\partial t^2} - c^2 \frac{\partial^2 \varphi}{\partial x^2} = 0, \quad (3.2a)$$

which is approximated by the standard second-order central-difference scheme

$$\frac{\varphi_j^{n+1} - 2\varphi_j^n + \varphi_j^{n-1}}{\tau^2} - c^2 \frac{\varphi_{j+1}^n - 2\varphi_j^n + \varphi_{j-1}^n}{h^2} = 0, \quad (3.2b)$$

constructed on the rectangular grid of variables  $x$  and  $t$  with sizes  $h$  and  $\tau = h/c$  respectively, using the five-node stencil shown in Figure 3.2. (Note, all the schemes used for simulations in Section 2.5, see (2.21), (2.22), and (2.23), are of the same central-difference explicit three-level type. This, however, is by no means a limitation — the ABCs can be constructed for any type of discretization.)

To create the discrete near-boundary sources similar to those of Section 3.1, and eventually set the discrete ABCs, we will need to be able to apply inside  $S$  the same finite-difference wave operator of (3.2b) as the one we are using on  $\mathbb{R} \setminus S$ . As such, we formally extend the exterior discretization, i.e., the rectangular grid with  $h \times \tau$  cells, into the interior domain  $S$ , as shown in Figure 3.2. We re-emphasize, however, that

<sup>6</sup>Generally speaking, one-dimensional problems do not have lacunae (except in special cases); as such, this example will only demonstrate the formal construction of the ABCs on the grid.

this is done only for the “artificial” purpose of building the ABCs. The actual governing equation in the near field, i.e., on  $S$ , as well as its discrete counterpart, may be more complex than the wave equations (3.2) above. In fact, neither the scheme stencil nor the grid used for computations in the near field have to be the same as those in the far field (although, they, of course, may). We only require that the exterior scheme (3.2b) with the stencil shown on Figure 3.2 be applicable till  $x = 0$ . And all we need to know from the standpoint of setting the ABCs, is that the overall combined problem (near field and far field) is uniquely solvable and well-posed.

Let us now consider all nodes of the aforementioned rectangular grid that belong to  $S$  (on all time levels), i.e., those for which  $x_j \equiv jh \leq 0$ . We denote this set of grid nodes by  $\mathcal{N}^+$ ; the complementary set that consists of all nodes from  $\mathbb{R} \setminus S$ , i.e., those, for which  $x_j > 0$ , is denoted by  $\mathcal{N}^-$ , see Figure 3.2. If we formally apply the five-node stencil of the scheme (3.2b) to each node from  $\mathcal{N}^+$ , then this stencil is obviously going to sweep one more vertical row of nodes, which already belongs to  $\mathcal{N}^-$  (i.e., to  $\mathbb{R} \setminus S$ ), and which is denoted by  $\gamma^-$  on Figure 3.2. Reciprocally, if we apply the stencil to every node in  $\mathcal{N}^-$ , it will also sweep the nodes  $\gamma^+$  that are already in  $\mathcal{N}^+$ . The two-layer grid structure  $\gamma = \gamma^+ \cup \gamma^-$  will be called the grid boundary; it represents on the discrete level the continuous interface between  $S$  and  $\mathbb{R} \setminus S$ , which is the vertical line  $x = 0$ .

Next, we assume that we integrate the interior problem one time step after another, and that we already know the discrete solution  $\varphi_S^{(h)}$  on the domain  $S$ , as well as the values of  $\varphi_\gamma^n$  on the grid boundary  $\gamma$ , up to a certain time level  $n$  (in particular,  $n$  may be equal to zero, which corresponds to the initial conditions).<sup>7</sup> These data obviously allow us to advance the next time step  $n + 1$  on  $\gamma^+$  and everywhere inside  $S$ ; in so doing, the outermost interior location  $\gamma^+$  on the level  $n + 1$  is computed by scheme (3.2b) using the stencil shown in Figure 3.2. These data, however, already do not allow us to calculate the discrete solution  $\varphi^{(h)}$  at  $\gamma^-$  on the level  $n + 1$ . And if this value  $\varphi_{\gamma^-}^{n+1}$  is not known, then we cannot advance further to level  $n + 2$ . Therefore, we conclude that the function of the ABCs in the discrete framework will be to provide the missing boundary values of the solution at  $\gamma^-$  on all time levels, one after another, starting from  $n = 1$ . This indeed constitutes the closure of the discrete system solved on  $S$ .

To provide the foregoing missing boundary value  $\varphi_{\gamma^-}^{n+1}$  for a given  $n$ , we recall that even so we do not know the discrete solution on level  $n + 1$  beyond  $\gamma^+$  (i.e., we do not know  $\varphi_{\mathcal{N}^-}^{n+1}$ ), we do know that the solution  $\varphi_S^{(h)}$  can be complemented on  $\mathcal{N}^-$  to a solution of equation (3.2b) on all time levels till  $n + 1$ . For our purposes, we will only need the existence of this complement rather than its actual representation. Let us now introduce a multiplier function  $\mu$  similar to that we have used in Section 3.1. The near-boundary interior transition region for this multiplier is schematically shown by the shaded area on Figure 3.2. We apply this multiplier to the combined discrete solution  $\varphi_c^{(h)} \equiv \varphi_{\mathcal{N}^+ \cup \mathcal{N}^-}^{(h)}$  on all time levels including  $n + 1$  (obtaining  $\varphi_{\mathcal{N}^+}^{(h)}$  may require projecting  $\varphi_S^{(h)}$  onto the grid  $\mathcal{N}^+$ ). In so doing, nothing changes on  $\mathcal{N}^- \cup \gamma^+$ , because  $\mu \equiv 1$  for  $x \geq 0$ . All the changes due to multiplication of  $\varphi_c^{(h)}$  by  $\mu$  will obviously be introduced on  $\mathcal{N}^+ \setminus \gamma^+$  only. Those amount to a smooth passage within the transition region (see Figure 3.2) from the actual unaltered values of the solution on  $\gamma^+$  to zero “well inside” the computational domain.

Next, similarly to Section 3.1, we apply the discrete wave operator  $\square^{(h)}$  of (3.2b) to the modified solution  $\mu\varphi_c^{(h)}$ . As  $\varphi_c^{(h)}$  is defined up to the level  $n + 1$ , the result  $g^{(h)}$  will be defined up to the level  $n$ . Analogously

<sup>7</sup> We use the subscript “ $S$ ” in  $\varphi_S^{(h)}$  rather than “ $\mathcal{N}^+$ ” to emphasize that the actual interior discrete solution may be computed on a different grid.

to (3.1), we obtain for all levels till  $n$ :

$$\square^{(h)}(\mu\varphi_c^{(h)}) = g^{(h)} \begin{cases} = 0 & \text{on the grid } \mathcal{N}^- \\ \neq 0 & \text{on } \gamma^+ \text{ \& in the transition region} \\ = 0 & \text{"well inside" } \mathcal{N}^+ \end{cases} \quad (3.3)$$

Notice, we can claim that the result in (3.3) is zero on  $\mathcal{N}^-$  only because nothing has been modified by  $\mu$  on  $\gamma^+$  and beyond. As such, we are simply using the fact that  $\varphi_c^{(h)}$  is a solution to the homogeneous equation outside the computational domain, and we do not need to know the explicit form of this solution.

Finally, we solve the non-homogeneous counterpart to equation (3.2b) driven by the RHS  $g^{(h)}$  of (3.3) everywhere on  $\mathcal{N}^+ \cup \mathcal{N}^-$ ; this will be henceforth referred to as solving the auxiliary problem. According to our construction, solving the auxiliary problem will allow us, in particular, to recover the value of  $\varphi_{\gamma^-}^{n+1}$ , which was not known previously, and which can now be supplied to the interior scheme as the missing boundary value. This means that we will have provided the ABCs for the interior problem, because in so doing we complete the time level  $n + 1$  and facilitate advancing the next level  $n + 2$ .

There are several important comments to be made regarding the foregoing ABCs' algorithm. At a first glance, the new formulation simply does not change much from the standpoint of solving the original infinite-domain problem. Indeed, all we have done is replaced the interior problem by the artificial near-boundary sources so that the exterior solution remain unaffected. Then, we suggested to use this exterior solution to close the interior discretization. However, obtaining this solution, i.e., solving the auxiliary problem, basically brings along the exact same set of complications that we have been trying to avoid by introducing the ABCs. Indeed, the domain of the auxiliary problem  $\mathcal{N}^+ \cup \mathcal{N}^-$  is still unbounded and as such, special treatment will be required for its numerical solution.

There is, however, a fundamental difference. The new auxiliary problem is linear throughout the entire space, and it is driven by known sources that are compactly supported inside the computational domain  $S$ . Consequently, the lacunae-based algorithm of Section 2 appears to be a most natural tool to solve it.<sup>8</sup> Employing the lacunae-based algorithm immediately implies that the domain of the auxiliary problem becomes bounded. Moreover, the "sufficiently retarded" sources do not contribute to its solution (see Section 2), i.e., only a limited extent of temporal pre-history of the solution will be needed to sustain the continuous time marching no matter how far in time we would like to advance the solution. In other words, the missing boundary value for the interior discretization  $\varphi_{\gamma^-}^{n+1}$  can be obtained using only finite computer resources in terms of both memory and number of arithmetic operations. Furthermore, these resources (say, per time level) will not increase no matter for how long we may need to run the computation, i.e., how large  $n$  may become. In this sense, the proposed ABCs become "true ABCs," i.e., the procedure that guarantees the appropriate closure of the truncated problem with only finite non-growing amount of computer resources required. In addition to that, we are guaranteed that the ABCs as a part of the overall algorithm will not contribute toward the buildup of numerical error during long runs.

The proposed ABCs can obviously be implemented via alternating interior/exterior steps. Namely, we advance one time step in the interior (including  $\gamma^+$ ) assuming that all the data that we need from the previous time levels are available. The resulting newly calculated time level will be the only one to which the multiplier has not been applied yet. We multiply it by  $\mu$ , and then apply the direct operator thus obtaining

<sup>8</sup>We reiterate that this algorithm cannot be applied in the case of one space dimension, but our ultimate goal is three-dimensional problems, see Section 3.3, and the considerations of the current section are for illustrative purposes only.

the right-hand side  $g^{(h)}$ , see (3.3), on one more time level as well. Finally, we perform one step of the lacunae-based integration of the auxiliary problem driven by  $g^{(h)}$  and obtain the missing boundary value for the interior problem. Then, the procedure cyclically repeats itself. Summarizing, we can say that having advanced the interior solution, we then generate a new contribution to the RHS of the auxiliary problem and subsequently advance its solution, which, in turn, allows us to calculate the next interior step.

An important observation, which is easy to make, is that the missing boundary value  $\varphi_{\gamma^-}^{n+1}$  that the ABCs provide does not, of course, depend on the actual shape of the multiplier  $\mu$  in the transition region. Indeed,  $\mu$  is defined so that it does not alter the solution on the grid boundary  $\gamma$ . Consequently, when we first apply the direct operator to  $\mu\varphi_c^{(h)}$ , see (3.3), and then integrate the non-homogeneous wave equation driven by  $g^{(h)}$ , the solution on  $\gamma$  will remain unchanged no matter what changes have been introduced by  $\mu$  in the interior. As such, the value  $\varphi_{\gamma^-}^{n+1}$  will only depend on the values of the solution on  $\gamma$  on all previous time levels, as well as on  $\varphi_{\gamma^+}^{n+1}$ . Moreover, since all the operations that we perform when constructing the ABCs are linear, we can symbolically write the resulting boundary condition as a linear form:

$$\varphi_{\gamma^-}^{n+1} = \mathbf{l}(\varphi_{\gamma^+}^{n+1}, \varphi_{\gamma}^n, \varphi_{\gamma}^{n-1}, \dots). \quad (3.4)$$

Technically, the dependency of  $\varphi_{\gamma^-}^{n+1}$  on the previous time levels, see (3.4), involves all of the latter, from  $\varphi_{\gamma}^n$  all the way back till  $n = 0$ . However, the use of the lacunae in three space dimensions will allow us to truncate (3.4) and leave only several levels that immediately precede  $n + 1$ ; the number of the levels involved will be fixed and will not increase with the increase of  $n$ . As such, temporal non-locality of the ABCs will be limited, and this will not be a consequence of any approximation, but rather an implication of the fundamental properties of the problem. We also note that the representation of the ABCs in the form (3.4) serves primarily the reason of convenience and compactness in notations. In fact, the coefficients of the linear form  $\mathbf{l}$  never need to be known explicitly except, possibly, when multiple interior problems are solved with the same exterior model (means same grid, same geometry, same scheme). In this case it may be beneficial to calculate the form  $\mathbf{l}$  once and ahead of time, compared to the straightforward calculation of  $\varphi_{\gamma^-}^{n+1}$  many times according to the procedure outlined above.

It is also important to mention that boundary condition (3.4) can be obtained in the framework of a general unsteady ABCs' methodology proposed by Ryaben'kii in [24] (see also older work [26]) for a variety of problems, including multidimensional cases, domains of varying shape, and different types of schemes — explicit as well as implicit. Work [24, 26] describes the theoretical construction of the ABCs per se, and does not address any issues related to the actual computations (for example, using lacunae-based integration, as proposed in the current paper). The methodology of [24, 26] relies on the concepts of generalized potentials and boundary projection operators of Calderon's type obtained and implemented in the discrete framework by means of the difference potentials method, see [27–29]. In this perspective, the ABCs of [24, 26], and boundary condition (3.4) in particular, can be interpreted as discrete counterparts to Calderon's boundary equations with projections in the unsteady case. In the following Section 3.3 we will describe a direct approach to obtaining multidimensional ABCs on moving boundaries, with no explicit use of the apparatus of Calderon's projections, and will also show how to apply the lacunae-based algorithm to perform the computations needed for these boundary conditions.

To conclude this section, we emphasize that even so  $\varphi_{\gamma^-}^{n+1}$  obtained according to (3.4) formally does not depend on the shape of the multiplier  $\mu$  inside  $S$ , we still need to have this multiplier smooth. In other words, we could not have used, e.g., a step function, instead of  $\mu$ . The reason is that non-smoothness will ruin lacunae in the discrete solution (see Section 2 and [23] for more detail) and consequently, we will no



longer be able to use the lacunae-based integration for solving the auxiliary problem and as such, setting the ABCs.

**3.3. General Construction of Discrete ABCs.** Similarly to the setup of Section 2, we now consider a domain  $S(t) \in \mathbb{R}^3$  that has finite diameter  $d$  for all times, and other than that may travel in space according to a prescribed law with the only limitation that its maximum speed be subsonic:  $k < c$ .  $S(t)$  will be the computational domain, or near field. In the far field, i.e., outside  $S(t)$ , we assume that our model is governed by the homogeneous wave equation:

$$\frac{\partial^2 \varphi}{\partial t^2} - c^2 \left( \frac{\partial^2 \varphi}{\partial x_1^2} + \frac{\partial^2 \varphi}{\partial x_2^2} + \frac{\partial^2 \varphi}{\partial x_3^2} \right) = 0, \quad t \geq 0. \quad (3.5)$$

As we have discussed, inside  $S(t)$  the solution  $\varphi = \varphi(\mathbf{x}, t)$  may be governed by a more complex equation/system, but all we need to assume is that the overall problem be uniquely solvable and well posed under the condition of waves' radiation toward infinity. For simplicity, we also assume homogeneity of the initial data everywhere, which is, however, not a limitation (see [23]).

Let us now introduce the discretization grid for equation (3.5). In principle, we need this grid only in the far field, i.e., outside  $S(t)$ , because the interior problem may be discretized in a different way, as indicated before. As, however, we have also seen, to obtain the ABCs we need to set up the auxiliary problem for the non-homogeneous counterpart of equation (3.5) driven by the special near-boundary sources. The auxiliary problem is to be formulated and solved on the entire space. As such, we introduce the grid for the linear wave equation on the entire  $\mathbb{R}^3 \times [0, \infty)$  as well. We denote by  $\mathcal{N}$  the collection of all grid nodes in  $\mathbb{R}^3 \times [0, \infty)$ , on which we evaluate the solution  $\varphi$ . Since (3.5) is an evolution equation, it is convenient to consider  $\mathcal{N}$  as a composition of spatially aligned grid hyper-planes:  $\mathcal{N} = \mathcal{N}_0 \cup \mathcal{N}_1 \cup \dots \cup \mathcal{N}_n \dots$ . Each  $\mathcal{N}_n$  is a spatial grid on  $\mathbb{R}^3$ , and we emphasize that they may, but do not have to, be the same on different levels  $n$ .

Let the individual nodes of the grid  $\mathcal{N}$  be denoted by  $\mathbf{n}$ . Equation (3.5) is approximated by a finite-difference scheme, which we assume, of course, to be consistent and stable:

$$\sum_{\mathbf{n} \in \mathcal{N}_m} a_{m\mathbf{n}} \varphi_{\mathbf{n}} = 0. \quad (3.6)$$

In (3.6),  $\mathcal{N}_m$  denotes the stencil attributed to the node  $\mathbf{m}$ , and  $a_{m\mathbf{n}}$  are the corresponding coefficients. When we say that the stencil is attributed to a particular node, we mean that the residuals of the discrete equation are evaluated at this particular grid location. In regard to this, we note that the residuals of the discretized equation (3.5) may, but do not have to, be evaluated on the same grid  $\mathcal{N}$ . To preserve the generality of the discussion, we assume that there is another, different, grid  $\mathcal{M}$  in  $\mathbb{R}^3 \times [0, \infty)$ , on which we keep the residuals, as well as the right-hand sides, if any, of the discrete wave equation. The subscript  $\mathbf{m}$  in equation (3.6) basically refers to this grid:  $\mathbf{m} \in \mathcal{M}$ . In the one-dimensional example of Section 3.2, both grids  $\mathcal{N}$  and  $\mathcal{M}$  were simply the same, and we did not have to distinguish between the two. To give an example of the opposite type, we mention the Yee scheme, see [30], which is one of the primary tools for discretizing the Maxwell's equations,<sup>9</sup> and which involves staggering in both space and time.

Next, we introduce two subsets of nodes of the grid  $\mathcal{N}$ . Let the level  $\mathcal{N}_n$  correspond to the actual moment of time  $t_n$ . For every  $n$ , we define  $\mathcal{N}_n^+$  as the set of all those and only nodes on this level that belong to the domain  $S(t_n)$ , and  $\mathcal{N}_n^-$  as the complement of  $\mathcal{N}_n^+$  to the entire  $\mathcal{N}_n$ :  $\mathcal{N}_n^- = \mathcal{N}_n \setminus \mathcal{N}_n^+$ . In other words,  $\mathcal{N}_n^-$

<sup>9</sup>The simplest version of the Maxwell's equations describes the propagation of electromagnetic waves in vacuum, which is a wave model similar in many respects to that given by (3.5).

contains all those and only those nodes of  $\mathcal{N}_n$  that belong to  $\mathbb{R}^3 \setminus S(t_n)$ . Subsequently, we define the set  $\mathcal{N}^+$  as the composition of all  $\mathcal{N}_n^+$  for all levels, and the set  $\mathcal{N}^-$  as the composition of all  $\mathcal{N}_n^-$  for all levels:

$$\mathcal{N}^+ = \bigcup_n \mathcal{N}_n^+, \quad \mathcal{N}^- = \bigcup_n \mathcal{N}_n^-.$$

Clearly,  $\mathcal{N}^- = \mathcal{N} \setminus \mathcal{N}^+$ .

In our definition of the scheme, see (3.6), we have identified the stencil  $\mathcal{N}_m$  with the grid location  $m$ , at which the residual is evaluated. From here on, we will assume for simplicity that the scheme (3.6) is explicit. In this case, there is only one non-zero coefficient  $a_{mn}$  on the upper time level of the stencil  $\mathcal{N}_m$ . We will denote the corresponding grid node by  $\hat{n}$ , and when it may not cause confusion, we will refer to the same stencil as either  $\mathcal{N}_m$  or  $\mathcal{N}_{\hat{n}}$ . It will also be convenient to introduce the four-dimensional (space-time) vector  $b = \hat{n} - m$ . The meaning of this vector is that it defines the relative position of the node  $\hat{n}$ , at which the upper-level coefficient is non-zero:  $a_{m\hat{n}} \neq 0$ , with respect to the “center” of the stencil  $m$ . This vector is obviously constant, it depends only on the local structure of the stencil, and does not depend on where exactly on the grid this stencil is applied at every given moment. In the one-dimensional example considered previously, we would have  $b = (\tau, 0)$ .

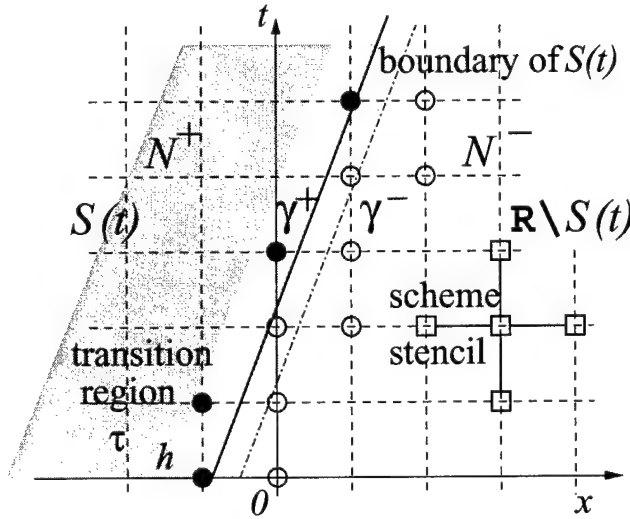


FIG. 3.3. One-dimensional illustration to the case of a moving domain.

are exactly those that need to be provided by the ABCs from the exterior side so that to be able to calculate the solution at every interior node  $n \in \mathcal{N}^+$  using the scheme (3.6). Reciprocally, the stencil  $\mathcal{N}_{\hat{n}}$  applied to every node  $\hat{n} \in \mathcal{N}^-$  sweeps additional nodes  $\gamma^+ \subset \mathcal{N}^+$ , see “bullets” on Figure 3.3:

$$\gamma^+ = \left[ \bigcup_{\hat{n} \in \mathcal{N}^-} \mathcal{N}_{\hat{n}} \right] \cap \mathcal{N}^+.$$

The set  $\gamma^+$  complements  $\gamma^-$  to the complete grid structure known as the grid boundary  $\gamma$  (see [28]):

$$\gamma = \gamma^+ \cup \gamma^-.$$

Let us now apply the stencil  $\mathcal{N}_{\hat{n}}$  to every node  $\hat{n} \in \mathcal{N}^+$ ; in so doing, the stencil obviously sweeps the entire grid  $\mathcal{N}^+$ , as well as a portion of the grid  $\mathcal{N}^-$  next to the interface, we will denote this portion by  $\gamma^-$ :

$$\gamma^- = \left[ \bigcup_{\hat{n} \in \mathcal{N}^+} \mathcal{N}_{\hat{n}} \right] \cap \mathcal{N}^-.$$

On Figure 3.3, we present a one-dimensional illustration<sup>10</sup> similar to that in Figure 3.2 for the case of a uniform motion of the computational domain, when the space-time trajectory of the boundary is a straight line (the set  $\gamma$  is denoted by small circles). From the standpoint of implementation, the values of the solution at the nodes  $\gamma^-$

<sup>10</sup>We note again that everywhere in this section the one-dimensional examples are for illustration purposes only, the actual algorithm is three-dimensional.

The grid boundary is a multi-layer fringe of nodes (two-layer fringe in the particular case of a second-order scheme with the stencil depicted in Figure 3.3) that is located near the continuous boundary and straddles it in some sense (cf. to Section 3.2).

To proceed with the construction of the ABCs, we will need to assume hereafter that the region of linearity, i.e., the area where the solution of the overall combined problem is governed by the linear homogeneous wave equation, extends “a little bit” to the interior of the computational domain  $S(t)$  as well. More precisely, this region will be assumed to extend inward at least as far as the entire grid boundary  $\gamma$ . This obviously presents no limitation from any standpoint. The multiplier  $\mu = \mu(\mathbf{x}, t)$  in this case is required to be identically equal to one,  $\mu(\mathbf{x}, t) \equiv 1$ , not only outside  $S(t)$ , but also inside — again, to the extent of  $\gamma^+$ . As such, the transition region for the multiplier, which is schematically shown by the darker gray shading on Figure 3.3, is shifted away from the boundary of  $S(t)$ .

To actually build the ABCs, we will perform the same procedure as outlined in Section 3.2. What we actually need in the discrete framework is to obtain the missing exterior boundary values of the solution  $\varphi_{\gamma^-}^{n+1}$  on every time level  $n + 1$ , or in other words, to complete this time level, so that to be able to advance the next time step. To do that, we take the solution already computed inside  $S(t)$  up to the level  $n + 1$ , multiply it by  $\mu$  and then apply the discrete operator  $\square^{(h)}$  everywhere. In doing so we assume, as mentioned before, that starting from  $\gamma$  outward, the solution satisfies the discrete homogeneous wave equation. In the general case that we are looking at now, an application of the operator  $\square^{(h)}$  brings us from the grid  $\mathcal{N}$  to the grid  $\mathcal{M}$ . The construction of the grid boundary  $\gamma$  and multiplier  $\mu$  guarantees that on all time levels up to  $n$  the near-boundary artificial sources  $g^{(h)}$  will satisfy (cf. to (3.3)):

$$\square^{(h)} \left( \mu \varphi_c^{(h)} \right) = g^{(h)} \begin{cases} = 0 & \text{for such } \mathbf{m} \in \mathcal{M} \text{ that } \mathbf{m} + \mathbf{b} \in \mathcal{N}^- \\ \neq 0 & \text{for such near-boundary } \mathbf{m} \in \mathcal{M} \text{ that } \mathbf{m} + \mathbf{b} \in \mathcal{N}^+ \\ = 0 & \text{one the grid } \mathcal{M} \text{ “well inside” } S(t) \end{cases} \quad (3.7)$$

Formula (3.7) suggests that in addition to the actual interface between the domains, i.e., the boundary of  $S(t)$ , it will also be convenient to consider another space-time trajectory obtained from this original interface by the constant displacement  $-\mathbf{b}$ , it is shown by the dash-dotted line in Figure 3.3. The right-hand side  $g^{(h)}$  will be zero on  $\mathcal{M}$  everywhere outside this new displaced boundary, and will differ from zero right next to it on the interior side. On Figure 3.3, we schematically show by the lighter gray shading the region where we still have  $\mu = 1$  but the RHS  $g^{(h)}$  may already differ from zero. Note, in the example of Section 3.2 we did not need to consider the displaced interface because the domain  $S(t)$  was stationary and the displacement  $\mathbf{b}$  was parallel to the time axis:  $\mathbf{b} = (\tau, 0)$ . Another important case when the two boundaries would coincide is  $\hat{\mathbf{n}} = \mathbf{m} - \mathbf{b} = \mathbf{0}$ . However, we cannot generally assume that for time-dependent problems. On the other hand, we mention that the grid boundaries originally introduced in [27, 28] and previous publications by Ryaben’kii for the solution of steady-state problems using difference potentials method, have been constructed so that just the center  $\mathbf{m}$  of the stencil  $\mathcal{N}_m$  (where the residuals are evaluated) would sweep a given grid subdomain and as such, generate the aforementioned fringe of nodes  $\gamma$  next to the continuous boundary.

Let us now make a few remarks of explanatory nature regarding the structure of the grid boundary  $\gamma$ . It obviously depends only on the type of the stencil  $\mathcal{N}_m$  and geometry of the actual continuous boundary that it straddles. From the definition of  $\gamma$  it is easy to see that once we have a solution to the homogeneous equation on  $\gamma$  and everywhere in the exterior, and operate by  $\square^{(h)}$  on this solution, then we can guarantee without actual calculation that the result will be zero for all  $\mathbf{m} \in \mathcal{M}$ :  $\mathbf{m} + \mathbf{b} \in \mathcal{N}^-$ . However, we cannot

“touch” even one single node from  $\gamma$  so that not to lose this property. If, for example, we allow for an alteration (via  $\mu \neq 1$ ) of a node from  $\gamma^+$  (see Figure 3.3), it will necessarily affect the exterior RHS. The latter may, generally speaking, become non-zero at some node(s)  $\mathbf{m} \in \mathcal{M}$ :  $\mathbf{m} + \mathbf{b} \in \mathcal{N}^-$ , and we will no longer be able to actually calculate it because we do not know anything about the exterior solution beyond  $\gamma$  except that it satisfies the homogeneous equation. We see, therefore, that it would ruin the entire derivation. On the other hand, the construction of the grid boundary  $\gamma$  is consistent in the sense that to calculate the actual non-zero near boundary sources for those  $\mathbf{m} \in \mathcal{M}$ , for which  $\mathbf{m} + \mathbf{b} \in \mathcal{N}^+$ , it is sufficient to know the solution only on  $\gamma$  and further inward, nothing outside  $\gamma$  needs to be known. As for the values  $\varphi_{\gamma}^{(h)}$  that are still needed, those are provided by the ABCs’ algorithm on every time level and as such, are available on all subsequent levels for calculating the source terms  $g^{(h)}$  of the auxiliary problem.

Having outlined the construction of the grid boundary  $\gamma$  and near-boundary sources  $g^{(h)}$  in the general case, we build the actual ABCs’ algorithm in much the same way as described previously. We perform the alternating interior/exterior steps: First advance one step in the interior, then apply  $\mu$  and calculate one more level of the sources  $g^{(h)}$ , and finally make one step of the lacunae-based integration of the auxiliary problem driven by these sources (see Section 2), thereby providing the missing data for advancing the next interior step. Then, the procedure cyclically repeats itself. To solve the auxiliary problem in this general case, we will obviously need a full-fledged version of the lacunae-based algorithm (see Section 2) that accounts for the motion of the sources and employs periodic boundary conditions in space and continuous time-marching with cyclic subtraction of the retarded contributions. As has been shown, implementation of the lacunae-based integration technique guarantees that the domain of the auxiliary problem will be bounded, and the computer resources needed for the ABCs will be finite and will not grow with time. Other properties of the ABCs outlined in Section 3.2, namely, independency on the shape of the multiplier  $\mu$ , and possibility to express the boundary values on the current time level as a linear function of the values on the previous levels, see (3.4), hold in the general framework of this section as well. Because of the lacunae, the aforementioned linear form will depend only on the finite non-increasing number of the preceding time levels  $n$ , essentially those included in the summation (2.11) once this formula is discretized on the grid  $\mathcal{N}$ . This means that the temporal nonlocality of the ABCs will be limited. As for the multiplier  $\mu$ , it has to be chosen sufficiently smooth so that to maintain good quality of the lacunae in the discrete solution, see Section 4.

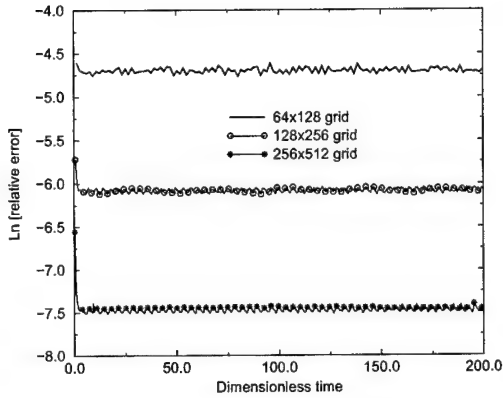
#### 4. Numerical Experiments with the ABCs.

**4.1. The Wave Equation with Known Exact Solution.** The first case that we analyze in the framework of the ABCs is actually the exact same problem that we solved in Section 2.5. It is the wave equation driven by a compactly supported oscillatory source in straightforward uniform motion. The exact solution of the problem was obtained using the Lorentz transform. The key difference between the current approach and that of Section 2.5 is that previously we have applied the lacunae-based algorithm directly to the original problem. Here, we rather decompose the problem into the near field  $S(t)$  and far field  $\mathbb{R}^3 \setminus S(t)$ , even so both are governed by the same wave equation. The integration in the near field is then performed by the conventional time marching. The exterior closure needed to sustain this time marching is provided by the discrete ABCs on the boundary of  $S(t)$ . The ABCs are constructed on the basis of the procedure outlined above — through the lacunae-based integration of the artificial near-boundary sources.

Similarly to Section 2.5, we have implemented three different schemes (2.21), (2.22), and (2.23), and every time integrated the problem till the dimensionless time has reached  $t = 200 \cdot d/c$ . The multiplier  $\mu = \mu(r, z, t)$  was constructed so that to have four continuous derivatives with respect to all its arguments.

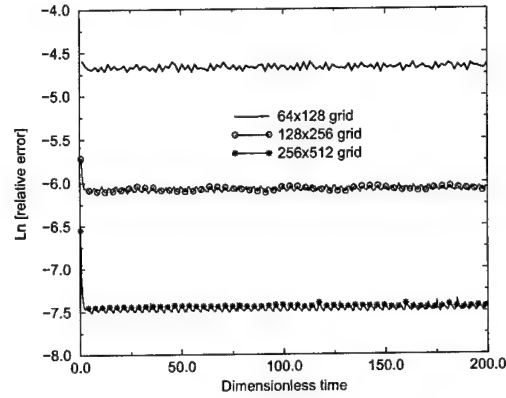
Smooth transition from 0 to 1 was obtained with the help of algebraic polynomials of degree 9 similar to those used in Section 2.5. The extent of the transition region has varied slightly between different cases with no noticeable effect on the quality of the solution. For all computational variants that we have considered it was within the range of several grid cells (typically, eight to ten, see Section 4.4 for further detail).

Lacunae-based ABCs for (2,2) cell-centered scheme



(A) The second-order scheme (2.21)

Lacunae-based ABCs for (2,2) node-centered scheme



(B) The second-order scheme (2.22)

FIG. 4.1. Grid convergence study with the lacunae-based ABCs for the wave equation.

Lacunae-based ABCs for (2,4) node-centered scheme

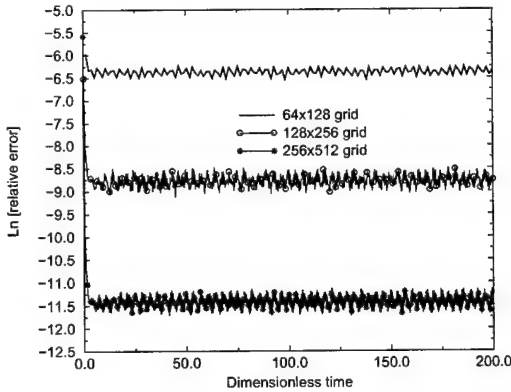


FIG. 4.2. Same as Fig. 4.1 for the fourth-order scheme (2.23).

the artificial near-boundary sources needed for constructing of the ABCs, rather than the original right-hand side  $f$  that drives equation (2.15).

An obvious observation which is easy to make is that Figures 4.1(A), 4.1(B), and 4.2, look practically indistinguishable from Figures 2.2(A), 2.2(B), and 2.3, respectively. In other words, the actual levels of the error on corresponding grids are essentially the same. As such, we conclude that in this most simple case the

In Figures 4.1 and 4.2 we present the results of the grid convergence study for the wave equation integrated with the lacunae-based ABCs over a long time interval. The errors are evaluated on the interior domain  $S(t)$  in the maximum norm. Computational setup completely follows that of Section 2.5, where we applied the lacunae-based algorithm to the original problem directly, see Figures 2.2 and 2.3. Namely, each scheme was implemented on a sequence of three grids:  $64 \times 128$ ,  $128 \times 256$ , and  $256 \times 512$ ; all geometric parameters, grid sizes, parameters that control the partition (2.7), as well as the actual exact solution (2.19), (2.20), against which we compare our numerical results, were taken exactly the same as before. We only note that in this case partition (2.7) applies to

introduction of the ABCs makes the outer boundary of the computational domain completely transparent for all the outgoing waves. This is equivalent to saying that the external boundary generates no reflection or alternatively, that any imperfections associated with the treatment of the outer boundary can always be kept on or below the level of the truncation error pertinent to the interior discretization. In this sense, the discrete lacunae-based ABCs that we have constructed can be regarded as an ideal closure of the interior finite-difference scheme. Experimentally, this is corroborated by the fact of non-deteriorating convergence of the scheme with the theoretically prescribed rate to the specially constructed exact solution of wave-radiation type on the computational domain  $S(t)$ .

**4.2. Nonuniform Motion of the Source.** The exact solution (2.19) of the wave equation driven by a point source in straightforward uniform motion was obtained in Section 2.5 using Lorentz' transform. The same solution could, of course, have been obtained by directly applying the Kirchhoff integral (2.3). The integration would require calculating explicitly the location, at which the trajectory of the source intersects the characteristic cone, and would lead to the same analytic result through a somewhat more complex derivation. In this section, we consider a somewhat more complex case of straightforward nonuniform (i.e., accelerated) motion of the source.<sup>11</sup> The Lorentz transform will obviously not apply in this case, but the Kirchhoff integral can still be used for obtaining the exact solution. However, for the general accelerated motion it may be impossible to analytically find the intersection of the characteristic cone with the source trajectory. Basically, it will require numerical computation, thus making the resulting exact solution only "semi-analytic." As such, to analyze the case of the accelerated motion, we have rather chosen a full-fledged numerical approach. First we calculate a fine-grid reference solution using the original lacunae-based algorithm of Section 2, and then compare against it the solutions obtained with the ABCs on coarser grids.

Taking the value of the speed of sound to be equal to one,  $c = 1$ , we consider the following law of accelerated motion for the center of the domain  $S(t)$ :

$$r = 0, \quad z \equiv z_0(t) = kt + k(\cos t - 1),$$

where  $k = 0.1$ . Keeping the values of all geometric parameters (sizes of the domains, etc.) the same as before, see Section 2.5, we introduce the following excitation for the wave equation (2.15):

$$f(r, z, t) = \cos\left(\frac{2\tilde{r}}{\kappa d} \frac{3\pi}{2}\right) \cdot P\left(\frac{2\tilde{r}}{\kappa d}\right) \cdot \left(1 + \frac{1}{4} \sin(\sqrt{2}t)\right) \cdot P\left(1 - \frac{t}{2\pi}\right), \quad (4.1)$$

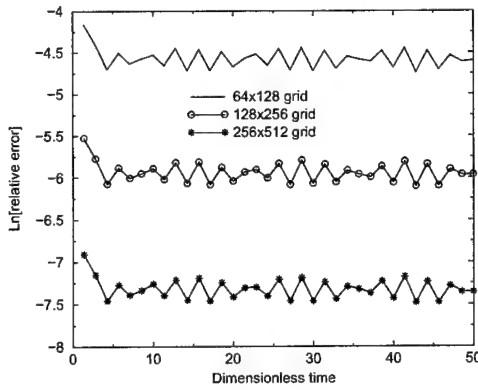
where  $d = 1.8$ ,  $\kappa = 0.8$ ,  $\tilde{r}^2 = r^2 + (z - z_0(t))^2$ , and  $P(\cdot)$  is the polynomial of 9th degree that decays smoothly from 1 to 0 on the interval  $[0, 1]$  so that  $P^{(m)}(0) = P^{(m)}(1) = 0$  for  $m = 1, 2, 3$ , and 4 (see Section 2.5). The function  $f$  of (4.1) obviously has four continuous derivatives everywhere with respect to all its arguments. Notice, the temporal behavior of this source function has been purposely chosen sufficiently complex; the frequency of the magnitude oscillations and that associated with the motion are incommensurable.

Equation (2.15) driven by the RHS (4.1) was integrated on the fine grid of dimension  $512 \times 1024$  till  $t = 50 \cdot d/c$  using the lacunae-based algorithm of Section 2 implemented with the fourth-order scheme (2.23). We have chosen here a shorter time interval compared to those we have used for previous demonstrations (Sections 2.5 and 4.1) so that not to make the computation of the reference solution excessively expensive. This interval is still quite sufficient for experimentally judging the convergence, see Figure 4.3 below. Having

<sup>11</sup>In all numerical examples we consider only straightforward motion because its direction has to be aligned with the  $z$  direction of the cylindrical coordinate system; otherwise, the symmetry will be lost. In general, this is not a limitation.

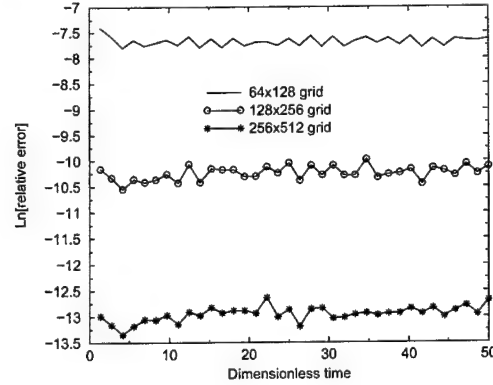
computed the fine-grid reference solution, we then integrated the same problem on a collection of coarser grids using the ABCs and compared the results with this reference solution. In so doing, we have employed only the two node-centered schemes: The second-order scheme (2.22) and the fourth-order scheme (2.23). The reason is that when both the fine-grid solution and the coarse-grid solution are calculated using a node-centered scheme, it is very easy to compare them point-wise (e.g., taking every other, every fourth, etc., node of the fine grid). In contradistinction to that, if we were to calculate a coarser-grid solution using the cell-centered scheme (2.21), then to compare it against the reference solution we would have had to use interpolation on the grid. This has a potential of contaminating the results because of the interpolation error, therefore we did not perform the aforementioned comparison for scheme (2.21).

Lacunae-based ABCs for (2,2) node-centered scheme



(A) The second-order scheme (2.22)

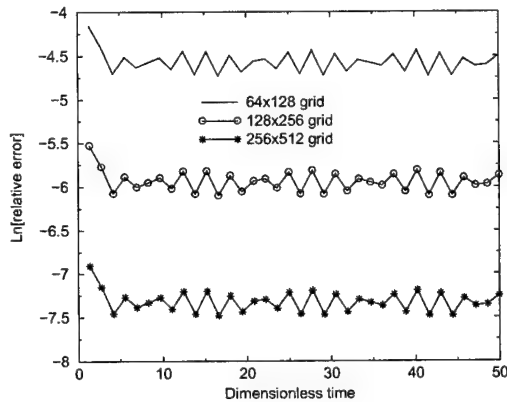
Lacunae-based ABCs for (2,4) node-centered scheme



(B) The fourth-order scheme (2.23)

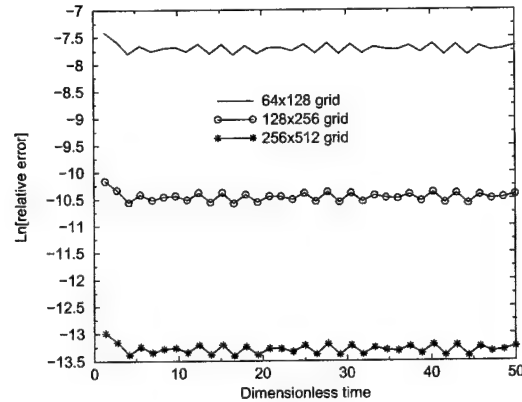
FIG. 4.3. Convergence of the solution of equation (2.15), (4.1) obtained with the ABCs to the fine-grid reference solution.

Lacunae-based (2,2) node-centered scheme



(A) The second-order scheme (2.22)

Lacunae-based (2,4) node-centered scheme



(B) The fourth-order scheme (2.23)

FIG. 4.4. Same as Figure 4.3, but the solution of (2.15), (4.1) is obtained with the original lacunae-based algorithm.



When computing the solution of equation (2.15) driven by non-uniformly moving source (4.1), we, of course, employ the computational domain  $S(t)$  that traces the motion of the source. As such, the ABCs are set on an artificial boundary that performs accelerated motion. On Figure 4.3, we compare the solutions obtained with the help of the lacunae-based ABCs with the fine-grid reference solution. Figure 4.3 shows the dependency of the numerical error on dimensionless time for different computational variants. As before, we use the sequence of three grids:  $64 \times 128$ ,  $128 \times 256$ , and  $256 \times 512$ , to demonstrate the convergence. Figure 4.3(A) clearly indicates the second-order convergence of scheme (2.22). For scheme (2.23), we observe the fourth-order convergence on Figure 4.3(B).

To assess the performance of the boundary conditions, we have computed the same solution on the same collection of coarser grids ( $64 \times 128$ ,  $128 \times 256$ , and  $256 \times 512$ ) with the same schemes (2.22) and (2.23), but with no ABCs, rather using the original lacunae-based algorithm of Section 2 (as we did when computing the reference solution). On Figure 4.4, we compare the results with the exact solution. As expected, scheme (2.22) converges uniformly in time with the second order (see Figure 4.4(A)), and scheme (2.23) — with the fourth order (see Figure 4.4(B)).

Comparing Figure 4.3(A) against Figure 4.4(A) we conclude that for the second-order scheme (2.22), the introduction of the ABCs again gives rise to no reflection back into the computational domain (i.e., no reflection beyond the level of the truncation error in the interior, cf. Section 4.1). As concerns the fourth-order scheme (2.23), one can still notice slight differences between the respective curves on Figures 4.3(B) and 4.4(B). The difference is most visible for the finest grid  $256 \times 512$ , less visible for the medium grid  $128 \times 256$ , and non-existent for the coarsest grid  $64 \times 128$ . This indicates that a small amount of reflections due to the ABCs may be present in the solution, although the actual elevation of the error on Figure 4.4(B) compared to 4.4(A) is so low that we can regard these reflections negligible anyway. Nonetheless, the discrepancy between the corresponding curves needs to be accounted for. We attribute it to the higher sensitivity of the fourth-order algorithm to the quality of the discrete lacunae. This phenomenon is commented on in Section 4.4; it is not of the fundamental nature, the quality of the lacunae can rather be controlled by appropriately choosing the parameters of the numerical procedure, more precisely, the shape and smoothness of the multiplier  $\mu$ .

Summarizing for the case of accelerated motion, we see that the solution of non-deteriorating quality on long time intervals can still be successfully computed using the lacunae-based ABCs. To the best of our knowledge, no other ABCs' methodology available in the literature can handle artificial boundaries of domains that move with acceleration while always keeping the reflections on or below the level of truncation error that pertains to a given interior discretization.

**4.3. Variable Speed of Sound.** For the last set of numerical experiments that we present in the paper, we wanted to pick up a case that would supposedly be more prone to the buildup of numerical error inside the computational domain  $S(t)$ . At the same time, we wanted to keep the computational setup “in line” with the previous experiments, see Sections 2.5, 4.1, and 4.2. In this connection, we notice that all the schemes that we have been using for numerical demonstrations previously were of explicit central-difference type. Consequently, the associated discretization error was of primarily dispersive nature. In the example of the current section, we will artificially increase the numerical dispersion inside the computational domain  $S(t)$  and experimentally assess the performance of the combined methodology (interior scheme and the ABCs) for this case. We should note, however, that the use of the ABCs is certainly not limited to the aforementioned class of the schemes.

It is known that numerical dispersion for central-difference schemes is more visible for more “suboptimal”

Courant numbers. In other words, the further below the stability limit the Courant number is, the more dispersive the numerical waves become. In particular, it is easy to see that the one-dimensional second-order scheme (3.2b) is exact, and simply reduces to pure propagation along the characteristics, when the Courant number  $\frac{\tau c}{h}$  is equal to 1. Reducing this number will introduce dispersion of numerical waves. (Of course, the convergence of the scheme still implies that the phase shift for every given frequency will become smaller as the grid sizes becomes smaller.) The analysis of the one-dimensional case also indicates that in multi-dimensional settings numerical dispersion is unavoidable. This is easy to understand already from the following qualitative consideration: To guarantee stability for all the waves propagating at an angle with respect to the grid lines one has to choose a smaller Courant number, which will necessarily appear suboptimal for those waves that propagate along the grid lines.

As has been mentioned, our intention now is to increase the numerical dispersion inside the computational domain and subsequently test the performance of the combined algorithm. To do that, we continuously reduce the speed of sound  $c$  from the peripheral areas of  $S(t)$  to its center. As stability across the entire domain will still be limited by the maximum speed of sound, the corresponding Courant number near the center will be suboptimal. This will imply higher levels of dispersion closer to the domain center. This will also mean that any perturbation that originates in  $S(t)$  will stay inside the domain longer compared to the previously analyzed cases of constant  $c$ . The explanation is obvious — the interior speed of propagation is lower. Consequently, we may expect that every particular wave will accumulate more error before it leaves the domain  $S(t)$ .

We emphasize that accurate quantification of the aforementioned phenomena is not of the central interest for discussion in the current paper. Therefore, we do not attempt to quantify the foregoing considerations, especially as it may appear technically difficult in any non-trivial setting. However, even on the level of qualitative understanding of the mechanisms of numerical dispersion, it is certainly of interest to experimentally assess the performance of the scheme with the lacunae-based ABCs for the case of variable  $c$ .

For our actual computations, we have chosen the following law of variation for the speed of sound inside the computational domain  $S(t)$ :

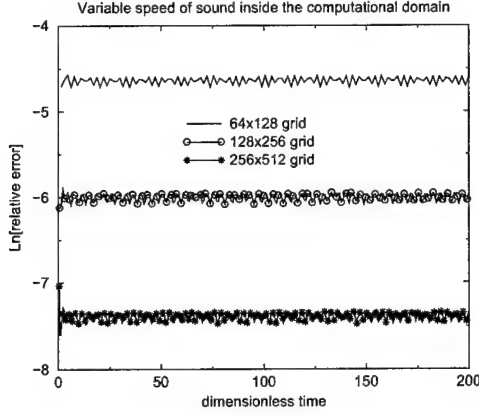
$$\tilde{c}^2 = c^2 \left( 1 - P_0 \cdot P \left( \frac{2\tilde{r}}{\kappa d} \right) \right), \quad (4.2)$$

where  $c$  is the original constant speed of sound in the far field,  $d$  denotes the diameter of the computational domain as before, and the polynomial  $P(\cdot)$ , as well as the quantities  $\tilde{r}$  and  $\kappa$ , have been introduced in Section 2.5. The constant  $P_0$  in expression (4.2) determines the extent of reduction in the speed of sound at the center of  $S(t)$ ; and we have tried two specific values:  $P_0 = 0.9$  and  $P_0 = 0.99$ , in our simulations. The solution that we have been computing in the case of variable speed of sound is the same traveling-source solution (2.19b), (2.20) that we analyzed before. However, instead of equation (2.15) we are now solving

$$\frac{\partial^2 \varphi}{\partial t^2} - \tilde{c}^2 \left( \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial \varphi}{\partial r} \right) + \frac{\partial^2 \varphi}{\partial z^2} \right) = f(r, z, t), \quad t \geq 0, \quad (4.3)$$

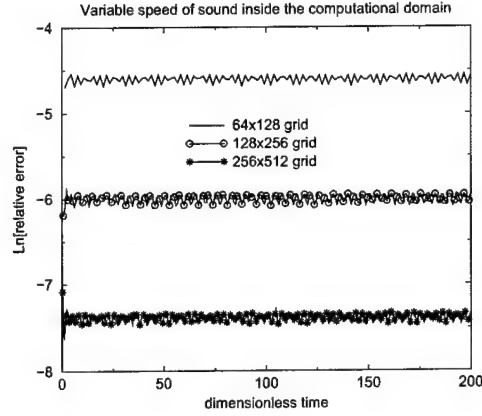
where  $\tilde{c}^2$  is defined by (4.2). Substituting  $\varphi(r, z, t)$  of (2.20) into the left hand side of equation (4.3) we obtain the source term  $f(r, z, t)$  that will obviously differ from that used previously in Sections 2.5 and 4.1. This new source term is still compactly supported on the domain  $S(t) \subset \mathbb{R}^3$  for all times, and it now drives the solution (2.20) on the entire space. As concerns the methodology for setting the ABCs, it remains exactly the same as before. Indeed, we point out that the variation of the speed of sound pertains to the original interior problem only. And the auxiliary problem that we solve for the purpose of setting the ABCs is by definition formulated with the constant speed of sound throughout its entire domain.

Lacunae-based ABCs for (2,2) cell-centered scheme



(A) The second-order scheme (2.21)

Lacunae-based ABCs for (2,2) node-centered scheme



(B) The second-order scheme (2.22)

FIG. 4.5. Convergence of the solution of equation (4.3), (4.2) with the ABCs to the exact solution (2.20) for  $P_0 = 0.9$ .

Lacunae-based ABCs for (2,4) node-centered scheme

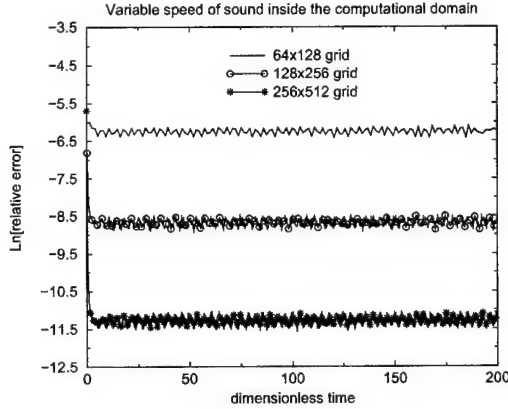


FIG. 4.6. Same as Fig. 4.5 for the fourth-order scheme (2.23).

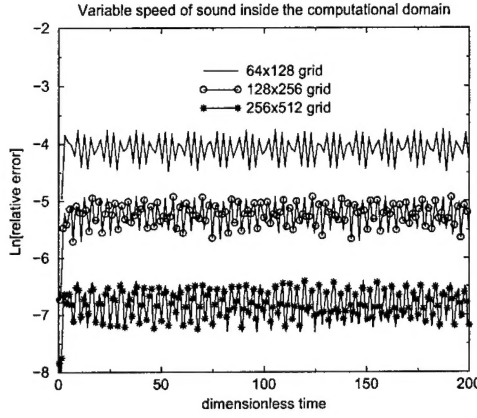
non-decreasing quality. Note, however, that as the original lacunae-based algorithm cannot be applied to the equation with variable  $c$ , we cannot directly compare in this case numerical results obtained with the ABCs against those obtained with no ABCs as we did before (see, e.g., comparison of the results on Figure 4.3 with those on Figure 4.4 in Section 4.2). As such, in making a conclusion that the ABCs in this case perform practically as well as they did in the previous cases, we rely on the foregoing experimental observation of temporally uniform convergence, as well as on the fact that the actual error levels on Figures 4.5 and 4.6 are only slightly higher than the respective levels on Figures 4.1 and 4.2. This is expected, because the results on Figures 4.1 and 4.2 correspond to numerically reproducing the exact same solution  $\varphi(r, z, t)$  of (2.20) using the ABCs but applying them to the original constant-coefficient wave equation in the interior.

Similar conclusions as to the convergence and quality of the numerical solution can be drawn for the case

In Figures 4.5 and 4.6 we present the results of the convergence study for the case  $P_0 = 0.9$  (see formula (4.2)) on the same sequence of three grids that we have used previously:  $64 \times 128$ ,  $128 \times 256$ , and  $256 \times 512$ . From Figures 4.5(A) and (B) we conclude, as before, that the algorithm converges with the second order for both schemes (2.21) and (2.22); and on Figure 4.6 we again observe the fourth-order convergence of scheme (2.23). The convergence obviously does not deteriorate as the time elapses, at least till dimensionless time reaches the moment  $200 \cdot d/c$  when we stop the computation. This shows that similarly to the previous cases of constant  $c$ , the proposed numerical procedure in the case of variable speed of sound is still capable of providing the solution of

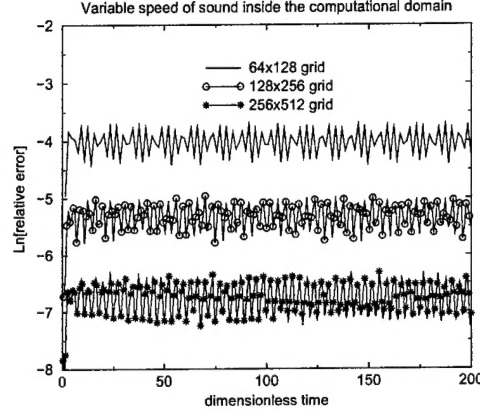
$P_0 = 0.99$  from looking at Figures 4.7(A) and (B) and Figure 4.8. From the qualitative considerations above it follows that this case is supposed to be “tougher” to compute, because the reduction in the speed of sound is more significant. In practice, this is manifested by noticeably more oscillatory error profiles, although we still clearly see that there is no deterioration of the solution in the long run. Besides, the actual levels of the error are somewhat higher compared to the corresponding curves on Figures 4.5(A) and (B) and Figure 4.6. This is also expected because the numerical dispersion inside  $S(t)$  is supposed to be higher as well.

Lacunae-based ABCs for (2,2) cell-centered scheme



(A) The second-order scheme (2.21)

Lacunae-based ABCs for (2,2) node-centered scheme



(B) The second-order scheme (2.22)

FIG. 4.7. Convergence of the solution of equation (4.3), (4.2) to the exact solution (2.20) for  $P_0 = 0.99$ .

Lacunae-based ABCs for (2,4) node-centered scheme

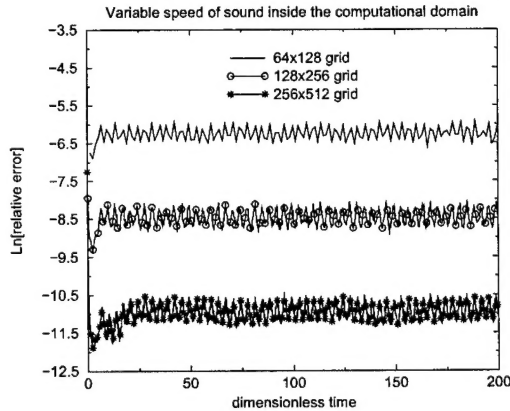


FIG. 4.8. Same as Fig. 4.7 for the fourth-order scheme (2.23).

uous solution will not be reproduced sufficiently accurately in the discrete solution of the auxiliary problem (essentially because the scheme will lose consistency, see discussion in the end of Section 3.2). In most of our computations we have used an algebraic polynomial function with four continuous derivatives for

**4.4. Implementation Notes.** The foregoing algorithm of lacunae-based ABCs has several parameters that need to be tuned appropriately so that to obtain the best possible results. Most of the flexibility associated with the algorithm resides in constructing the multipliers and artificial near-boundary sources needed for computing the ABCs (Section 3), as well as in choosing the parameters of the lacunae-based integration (Section 2). We have not yet conducted a comprehensive study of how the corresponding parameters affect the numerical procedure and as such, will only outline here some general trends.

As has been mentioned before, the multiplier has to be smooth in the transition region, see Figures 3.2 and 3.3. Otherwise, lacunae of the contin-

multiplier, and the extent of the transition region was about ten grid cells. This has always been sufficient for the second-order schemes. In other words, we could always obtain the temporally uniform second-order convergence with these settings, although we did not check, for example, whether or not it was possible to further reduce the extent of the transition region. As concerns the fourth-order convergence, we did see situations when the previous settings turned out somewhat insufficient (e.g., in Section 4.2). This occurred mostly when going from the medium grid  $128 \times 256$  to the finest grid  $256 \times 512$ . To maintain the convergence rate in this case, we had to use a wider transition region (fifteen cells) and/or a smoother multiplier (five continuous derivatives). This indicates that in general the algorithm of lacunae-based ABCs is sensitive to the smoothness of the multiplier, as it is supposed to be. However, this sensitivity does not actually manifest itself before the error reaches sufficiently low levels. As such, in practical computing one will most likely be able to use rather narrow transition regions, as well as multipliers with limited smoothness.

As concerns choosing the parameters of the lacunae-based integration (see Section 2), there is at least one important observation that has been made experimentally. In theory, the contribution of a given fragment of the RHS can be subtracted from the overall solution as soon as the time interval  $T_{\text{int}}$  has elapsed since its inception. In practice, it has been found useful to introduce the so-called aft front time gap  $\delta$ , i.e., allow for a little extra time for the waves to propagate outward. This implies choosing a somewhat larger value for the period  $Z$  compared to the necessary minimum given by (2.6) so that by the time of subtraction, which is  $T_{\text{int}} + \delta > T_{\text{int}}$ , the reflected waves will not have started re-entering the domain  $S(t)$  yet. Moreover, we can choose to introduce the actual front time gap as well, i.e., increase  $Z$  even further, so that by the time of subtraction of a given contribution the corresponding reflected waves still be at a (small) distance from  $S(t)$  rather than right next to its boundary. Experimentally, we have found that the aft front time gap affects the quality of the solution stronger than the actual front time gap. However, the quantity  $\delta$  in all our simulations was sufficiently small anyway, about 4% of  $T_{\text{int}}$ , and most likely it could be reduced even further.

**5. Conclusions.** We have constructed and tested the algorithm for setting highly-accurate global artificial boundary conditions in the problems of time-dependent wave propagation. The key building block of the new ABCs is a special non-deteriorating numerical procedure that has been developed previously for the long-term integration of wave-radiation problems. The latter procedure is based on the presence of lacunae (aft fronts of the waves) in the three-dimensional wave-type solutions. The resulting lacunae-based ABCs are obtained directly for the discrete formulation of the problem and can complement any consistent and stable finite-difference scheme. In so doing, neither a rational approximation of non-reflecting kernels, nor discretization of the continuous boundary conditions is required. The extent of temporal nonlocality of the new ABCs appears fixed and limited, and this is not a result of any approximation but rather a direct consequence of the fundamental properties of the solution. The proposed ABCs can handle artificial boundaries of irregular shape on regular grids with no fitting/adaptation needed. Besides, they possess a unique capability of being able to handle boundaries of moving computational domains, including the case of accelerated motion. We have conducted a series of numerical experiments that would corroborate the theoretical design properties of the algorithm. The experiments included computation of unsteady wave-radiation solutions over long time intervals. In all our experiments the ABCs could always keep the level of reflections from the artificial boundary on or below the level of truncation error for the interior discretization for as long as the computation was run. Besides the classical wave equation that we have analyzed in the current paper, the proposed technique may find applications in computational acoustics and computational electromagnetics.

## REFERENCES

- [1] D. GIVOLI, *Non-reflecting Boundary Conditions*, J. Comput. Phys., 94 (1991) pp. 1–29.
- [2] S. V. TSYNKOV, *Numerical Solution of Problems on Unbounded Domains. A Review*, Appl. Numer. Math., 27 (1998) pp. 465–532.
- [3] T. HAGSTROM, *Radiation Boundary Conditions for the Numerical Simulation of Waves*, Acta Numerica, A. Iserles, Ed., Vol. 8, Cambridge University Press, Cambridge, 1999, pp. 47–106.
- [4] J.-P. BERENGER, *A Perfectly Matched Layer for the Absorption of Electromagnetic Waves*, J. Comput. Phys., 114 (1994) pp. 185–200.
- [5] J.-P. BERENGER, *Three-Dimensional Perfectly Matched Layer for the Absorption of Electromagnetic Waves*, J. Comput. Phys., 127 (1996) pp. 363–379.
- [6] L. ZHAO AND A. C. CANGELLARIS, *GT-PML: Generalized Theory of Perfectly Matched Layers and its Application to the Reflectionless Truncation of Finite-Difference Time-Domain Grids*, IEEE Trans. Microwave Theory Tech., 44 (1996) pp. 2555–2563.
- [7] P. G. PETROPOULOS, L. ZHAO, AND A. C. CANGELLARIS, *A Reflectionless Sponge Layer Absorbing Boundary Condition for the Solution of Maxwell's Equations with High-Order Staggered Finite Difference Schemes*, J. Comput. Phys., 139 (1998) pp. 184–208.
- [8] S. ABARBANEL AND D. GOTTLIEB, *On the Construction and Analysis of Absorbing Layers in CEM*, Appl. Numer. Math., 27 (1998) pp. 331–340.
- [9] E. TURKEL AND A. YEFET, *Absorbing PML Boundary Layers for Wave-Like Equations*, Appl. Numer. Math., 27 (1998) pp. 533–557.
- [10] S. ABARBANEL AND D. GOTTLIEB, *A Mathematical Analysis of the PML Method*, J. Comput. Phys., 134 (1997) pp. 357–363.
- [11] S. ABARBANEL AND D. GOTTLIEB, *On the Construction and Analysis of the Absorbing Layers in CEM*, 13th Annual Review of Progress in Applied Computational Electromagnetics (1997) pp. 876–883.
- [12] M. J. GROTE AND J. B. KELLER, *On Nonreflecting Boundary Conditions*, J. Comput. Phys., 122 (1995) pp. 231–243.
- [13] M. J. GROTE AND J. B. KELLER, *Exact Nonreflecting Boundary Conditions for the Time-Dependent Wave Equation*, SIAM J. Appl. Math., 55 (1995) pp. 280–297.
- [14] M. J. GROTE AND J. B. KELLER, *Nonreflecting Boundary Conditions for Time-Dependent Scattering*, J. Comput. Phys., 127 (1996) pp. 52–65.
- [15] M. J. GROTE AND J. B. KELLER, *Nonreflecting Boundary Conditions for Maxwell's Equations*, J. Comput. Phys., 139 (1998) pp. 327–342.
- [16] I. L. SOFRONOV, *Artificial Boundary Conditions of Absolute Transparency for Two- and Three-Dimensional External Time-Dependent Scattering Problems*, European J. Appl. Math., 9 (1998) pp. 561–588.
- [17] I. L. SOFRONOV, *Non-Reflecting Inflow and Outflow in a Wind Tunnel for Transonic Time-Accurate Simulations*, J. Math. Anal. Appl., 221 (1998) pp. 92–115.
- [18] B. ALPERT, L. GREENGARD, AND T. HAGSTROM, *Rapid Evaluation of Nonreflecting Boundary Kernels for Time-Domain Wave Propagation*, SIAM J. Numer. Anal., 37 (2000) pp. 1138–1164.
- [19] B. ENGQUIST AND A. MAJDA, *Absorbing Boundary Conditions for the Numerical Simulation of Waves*, Math. Comput., 31 (1977) pp. 629–651.
- [20] B. ENGQUIST AND A. MAJDA, *Radiation Boundary Conditions for Acoustic and Elastic Wave Calculations*, J. Comput. Phys., 90 (1991) pp. 169–205.

- lations, Commun. Pure Appl. Math., 32 (1979) pp. 313–357.
- [21] B. ENGQUIST AND A. MAJDA, *Numerical Radiation Boundary Conditions for Unsteady Transonic Flow*, J. Comput. Phys., 40 (1981) pp. 91–103.
  - [22] V. S. RYABEN'KII, V. I. TURCHANINOV, AND S. V. TSYNKOV, *Long-Time Numerical Integration of the Three-Dimensional Wave Equation in the Vicinity of a Moving Source*, ICASE Report No. 99–23, NASA/CR–1999–209350, Hampton, VA, January 1999.
  - [23] V. S. RYABEN'KII, S. V. TSYNKOV, AND V. I. TURCHANINOV, *Long-Time Numerical Computation of Wave-Type Solutions Driven by Moving Sources*, to appear in Appl. Numer. Math.
  - [24] V. S. RYABEN'KII, *Nonreflecting Time-Dependent Boundary Conditions on Artificial Boundaries of Varying Location and Shape*, Appl. Numer. Math., 33 (2000) pp. 481–492.
  - [25] P. GARABEDIAN, *Partial Differential Equations*, Chelsea Pub. Co., New York, 1986.
  - [26] V. S. RYABEN'KII, *Exact Transfer of Difference Boundary Conditions*, Functional Anal. Appl., 24, No. 3 (1990) pp. 251–253.
  - [27] V. S. RYABEN'KII, *Boundary Equations with Projections*, Russian Math. Surveys, 40 (1985) pp. 147–183.
  - [28] V. S. RYABEN'KII, *Difference Potentials Method for Some Problems of Continuous Media Mechanics*, Nauka, Moscow, 1987. (in Russian)
  - [29] V. S. RYABEN'KII, *Difference Potentials Method and its Applications*, Math. Nachr., 177 (1996) pp. 251–264.
  - [30] K. S. YEE, *Numerical Solution of Initial Boundary Value Problem Involving Maxwell's Equations in Isotropic Media*, IEEE Trans. Antennas Propagat., 14 (1966) pp. 302–307.